
TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: B2612 – Elektrotechnika a informatika

Studijní obor: 3906T001 – Mechatronika

Řízení modelu přehřívání páry

Control of Steam Superheating Model

Diplomová práce

Autor: **Marek Hurt**

Vedoucí práce: Ing. Lukáš Hubka, Ph.D.

Konzultant: Ing. Petr Školník, Ph.D.

V Liberci 17. 5. 2013

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum

Podpis

Poděkování

Děkuji vedoucímu mé diplomové práce Ing. Lukášovi Hubkovi, Ph.D. za cenné rady, připomínky, odborné vedení práce, čas strávený při konzultacích a zajištění potřebného zázemí pro práci.

Abstrakt

Diplomová práce se zabývá řízením modelu přehřívání páry. Pro řízení bylo použito PLC od firmy Siemens, které je prostřednictvím OPC serveru od firmy Deltalogic propojeno s počítačem, ve kterém běží model navržený v prostředí Matlab.

V teoretické části je popsán model přehřívání páry a použité softwarové a hardwarové prostředky. Praktická část popisuje propojení OPC server – PLC a OPC server – Matlab. Následně se zabývá realizací jednotlivých dílčích celků potřebných k vytvoření řídicího obvodu v PLC dle technické dokumentace. Nakonec jsou prezentovány výsledky naměřené při finálním propojení modelu v Matlabu s řídicím systémem v PLC Siemens.

Klíčová slova

Přehřívání páry, PIL, HIL, OPC server, PLC Siemens, PI regulátor

Abstract

The diploma thesis deals with a steam superheating model control. A Siemens PLC was used for the process control. The PLC is connected with the model created in Matlab via an OPC server from DeltaLogic.

The theoretical part describes the steam superheating model and used software and hardware tools. The practical part describes a connection OPC server – PLC and OPC server – Matlab. After that, it deals with the realization of particular units needed for a creation of a control circuit in PLC which was created according to the technical documentation. The end of the thesis presents all results measured after the final model was connected with the control system in PLC Siemens.

Key words

Steam superheating, PIL, HIL, OPC server, PLC Siemens, PI controller

Obsah

Obsah	6
Seznam použitých obrázků	7
Seznam použitých zkratk	8
1 Úvod	9
2 Možné způsoby simulací	10
2.1 Simulace soustavy metodou „Model in the Loop“	10
2.2 Simulace soustavy metodami PIL a HIL	11
3 Popis modelované soustavy	13
3.1 Princip tepelné elektrárny	13
3.2 Model soustavy v Matlabu	14
4 PLC Siemens S7 – 300	15
4.1 Popis PLC Siemens S7 – 312C	15
4.2 Programování PLC Siemens S7 - 300	16
4.3 PI(D) regulátor	18
4.4 Základní pojmy používané při regulaci	19
5 OPC technologie	20
5.1 Specifikace OPC	20
5.2 Popis specifikace OPC Data Access	22
5.3 Výběr OPC serveru	22
5.4 OPC server Deltalogic	23
6 Nastavení komunikace mezi OPC serverem a klienty	24
6.1 Propojení OPC – Siemens	24
6.2 Propojení OPC server – Matlab	27
6.3 Ověření kompletní komunikace	28
7 Sestrojení regulátoru v PLC	35
7.1 Popis řídicího obvodu výstupního přehříváku	36
7.2 Testování dílčích částí realizace – funkční generátory	37
7.3 Testování dílčích částí realizace – filtry	39
7.4 Celkové zapojení a výsledky testování	40
8 Závěr	43
Seznam použité literatury	44
Seznam příloh	46

Seznam použitých obrázků

Obr. 1: Průběh návrhu modelu.....	10
Obr. 2: Simulace soustavy metodou Model in the Loop	11
Obr. 3: Typické schéma HIL systému (převzato z [5])	12
Obr. 4: Zjednodušené schéma výrobního bloku tepelné elektrárny (převzato z [6])	13
Obr. 5: Blokové schéma regulované VT části (převzato z [1])	14
Obr. 6: PLC Siemens S7 - 300.....	15
Obr. 7: Prostředí Step7.....	17
Obr. 8: Blokové schéma regulačního bloku FB41 (převzato z [7]).....	18
Obr. 9: Architektura OPC (převzato z OPC Foundation [9])	20
Obr. 10: Specifikace OPC a vazby mezi nimi	21
Obr. 11: Startovní okno OPC serveru Deltalogic	25
Obr. 12: Konfigurace OPC serveru - vytvoření nového připojení.....	25
Obr. 13: Konfigurace OPC serveru - konfigurace zařízení	26
Obr. 14: Bloky OPC Toolbox	27
Obr. 15: Nastavení parametrů OPC Toolboxu.....	28
Obr. 16: Ověření komunikace Matlab - PLC přes OPC	30
Obr. 17: Ověření komunikace Matlab - PLC přes OPC (Enabled Subsystem)	30
Obr. 18: Přechodové charakteristiky úlohy k ověření komunikace.....	31
Obr. 19: Porovnání výstupů „I“ složek regulátorů Matlab – PLC.....	31
Obr. 20: Výřez průběhů „I“ složek regulátorů Matlab – PLC (vzorkování 0,1s).....	32
Obr. 21: Porovnání různých výstupů "I" složky	32
Obr. 22: Simulace sinusového signálu.....	33
Obr. 23: Softing OPC Toolbox Demo Client - sledování proměnných na OPC serveru	34
Obr. 24: Řídicí obvod výstupního přehříváku	36
Obr. 25: Porovnání výstupů generátoru G6 v Matlabu a PLC.....	38
Obr. 26: Porovnání výstupů filtrů T=200 s v Matlabu a PLC (v běžící simulaci).....	40
Obr. 27: Řídicí část simulačního schéma s připojenou řídicí částí v PLC.....	41
Obr. 28: Výstup PI regulátoru vnější smyčky.....	42
Obr. 29: Výstup PI regulátoru vnitřní smyčky	42
Obr. 30: Blokové schéma regulačního bloku FB41[10]	51

Seznam použitých zkratk

COM	(Component Object Model) – komponentově orientovaná technologie od firmy Microsoft
DCOM	(Distributed COM) – distribuovaná verze technologie COM (COM + síťové rozšíření)
FBD	(Functional Block Diagram) – jazyk pracující s funkčními bloky, podobný LAD
HIL	Hardware in the Loop
LAD	(Ladder Diagram) – jazyk podobný reléovým schémátům
MIL	Model in the Loop
OLE	(Object Linking and Embedding) – technologie umožňující vkládání a propojování dokumentu a dalších objektů vyvinutá společností Microsoft
OPC	(OLE for Process Control) – standard vytvářející společné rozhraní pro nespočetné systémy
PIL	Processor in the Loop
PLC	(Programmable Logic Controller) – programovatelný logický automat
SCADA	(Supervisory Control and Data Acquisition) – dispečerské řízení a sběr dat
STL	(Statement List) – jazyk podobný Assembleru

1 Úvod

V praxi je velmi často výhodné vyzkoušet řídicí systém ještě před tím, než bude připojen k reálné řízené soustavě. K tomu se obvykle využívá metoda návrhového cyklu založená na modelu. Jako první se většinou navrhne systém, kdy jak model, tak regulátor jsou navrženy v nějakém simulačním prostředí, např. Matlab Simulinku. Tím se získá alespoň základní představa o tom, jak se bude daná soustava chovat. Nejsou zde však zachyceny další vlivy, kterými jsou např. přenosové rychlosti v reálné soustavě, či výkon použitého řídicího hardware. Proto je výhodné využít dalších možností simulace např. PIL a HIL, kdy se reálné soustavě přibližujeme již více. Tyto simulace jsou následně popsány v první kapitole.

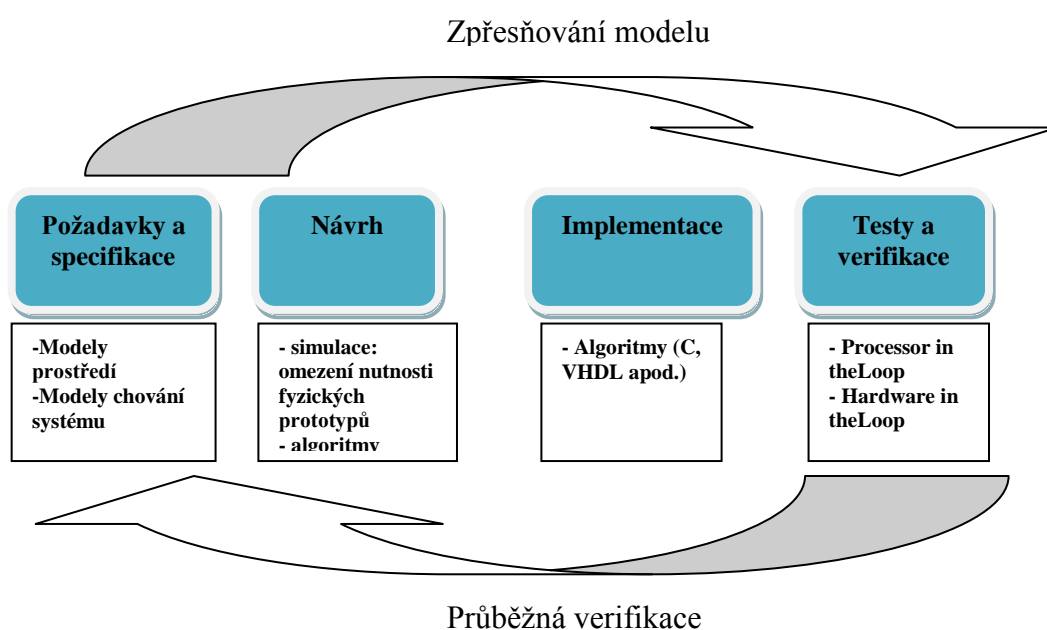
Úkolem diplomové práce je vytvořit dle dodané technické dokumentace řídicí obvod pro poslední technologický úsek tepelné elektrárny, kterým je výstupní přehřívání vysokotlaké páry.

Nejprve jsou teoreticky vysvětleny výhody a nevýhody jednotlivých metod simulace. Následně je popsán samotný řízený model přehřívání páry a důvody jeho vzniku [1]. Dále je také popsán použitý řídicí systém a nastíněny možnosti jeho programování. V neposlední řadě se teoretická část zabývá možnostmi a druhy OPC serverů, které se v praxi často používají. Podrobnější popis je věnován OPC serveru od společnosti Deltalogic, který je při práci použit.

V úvodu praktické části je věnována pozornost propojení OPC serveru jak s řídicím systémem Siemens, tak s Matlabem. V první aplikaci je toto propojení pracující v pseudo-reálném čase ověřeno a vyzkoušeno na logickém řízení a následně na řízení jednoduššího modelu realizovaného přenosovou funkcí. Poté je zrealizována již samotná část řízení vybraného úseku přehřívání páry. Nejprve po jednotlivých dílčích částech a následně jako celkové zapojení. Hlavní snahou je tedy převést řídicí část, která má standardní strukturu do použitého PLC. Bloková schémata zapojení regulačních obvodů jsou uvedena v příloze. Hlavním cílem je jejich naprogramování v PLC Siemens a propojení s modelem v Matlabu.

2 Možné způsoby simulací

Model-Based Design (MBD) je metoda vývojového cyklu zařízení, kde se předpokládá práce s matematickým modelem. Na začátku takového cyklu se nejprve specifikují požadavky na chování modelu. V dalším kroku se navrhují modely prostředí, komponent a řídicí algoritmy. Když jsou tyto navrženy, tak se implementují do použitých zařízení a na konec probíhají testy a verifikace. Tento proces se opakuje pořád dokola, dokud není dosaženo dostačujících výsledků. Na obr. 1 je tento cyklus graficky znázorněn.

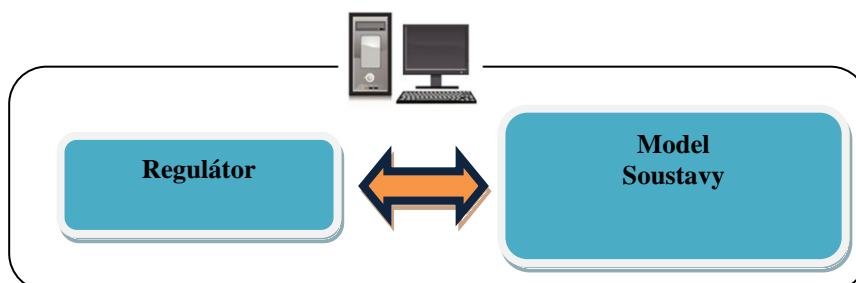


Obr. 1: Průběh návrhu modelu

2.1 Simulace soustavy metodou „Model in the Loop“

Vytvoření modelu je velmi důležitá část pro správné navrhování a odzkoušení funkčnosti regulačního obvodu. Ve většině případů řízení není možné při návrhu řídit přímo reálnou soustavu. Jedním z důvodů může být poškození této soustavy nesprávně zvoleným regulátorem. Dalším např. problém s připojením k soustavě, která je v provozu. V začátku návrhového cyklu je vhodné začít simulací metodou MIL. Při použití této metody je jak model soustavy, tak regulační část vytvořena v simulačním prostředí, např. v Matlab Simulinku. Na obr. 2 je zobrazeno schéma tohoto typu simulace. Pomocí uvedené metody získáme již

použitelné výsledky, i když v některých případech stále dost idealizované. Nevýhodou metody je, že s její pomocí nelze ověřit rychlosti komunikačních linek, ani výkonnost následně v praxi použitého hardware[2].



Obr. 2: Simulace soustavy metodou Model in the Loop

2.2 Simulace soustavy metodami PIL a HIL

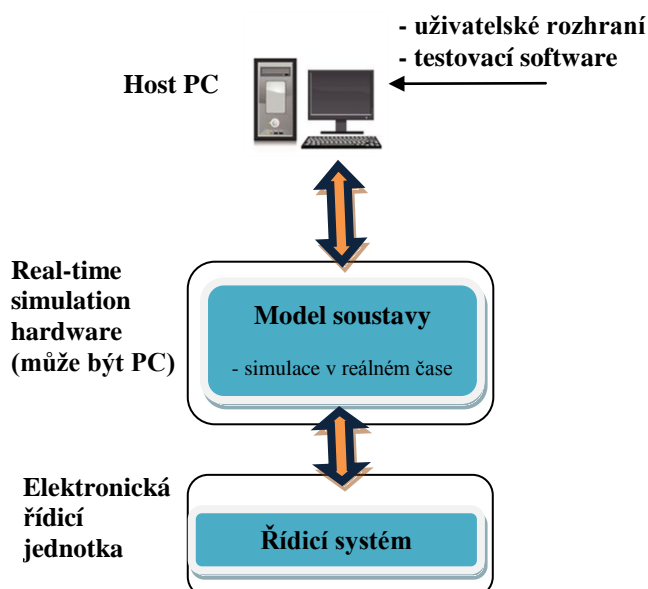
Dalším přiblížením reálné soustavy je využití simulace metodou PIL. Zde již neprobíhá kompletní simulace pouze v simulačním software v počítači, ale jsou vygenerovány zdrojové kódy, které se nahrají přímo do požadovaného řídicího hardware. Simulace potom může, ale nemusí probíhat v reálném čase.

Nejvýše postavená simulace je metoda HIL. Při této simulaci je řídicí obvod obsažen přímo ve stejném řídicím hardware, jaké bude ve skutečnosti použito u reálného procesu. I model bývá většinou nahrán do specifického simulačního hardware (nějakého embedded systému). Celá simulace probíhá v reálném čase a propojení mezi řídicí a řízenou částí je realizováno stejně, jako bude propojení v reálné soustavě. Takovýto systém poskytuje široké možnosti při odladování soustavy. Dovoluje odsimulovat i takové případy poruch, které by při připojení k reálné soustavě mohly skončit zničením soustavy [2], [3].

Protože simulace uvažovaná v této práci neběží přímo v reálném čase, ale využívá tzv. pseudo-reálného času (na straně OPC klienta v Matlabu, viz dále), bude se tato práce pokoušet o vytvoření simulace metodou PIL. I když označit ji již jako HIL by také nebylo úplně špatně.

Dříve, než se začaly používat tyto typy simulací, tak se např. v automobilovém průmyslu vyráběly prototypové modely motorů, či různých subsystémů a na nich se

potřebné testy prováděly. Vývoj těchto prototypů byl velmi nákladný a dlouhý. Často docházelo při testování i k jejich zničení. Zdroj [4] uvádí, že až 95% testů na novém automobilovém motoru bude prováděno metodou HIL. Výhodné využití je také při vyvíjení „fly-by-wire“ systémů pro řízení klapky na křídlech letadla pouze „po drátě“, bez mechanického spojení s kabinou. V tomto případě se použití HIL jeví jako velmi vhodné, protože se ušetří čas, kdy se mohou provádět testy ještě dříve, než je model nového letadla fyzikálně sestaven. Dále je zde také otázka bezpečnosti, protože se některé chyby mohou odhalit ještě na zemi a nedojde v extrémním případě až ke zničení letadla. Mezi takové testy patří také testování ABS systémů v krizových situacích nebo, jako v našem případě, testování části elektrárny [4]. Na níže uvedeném obr. 3 je znázorněno typické schéma HIL systému.

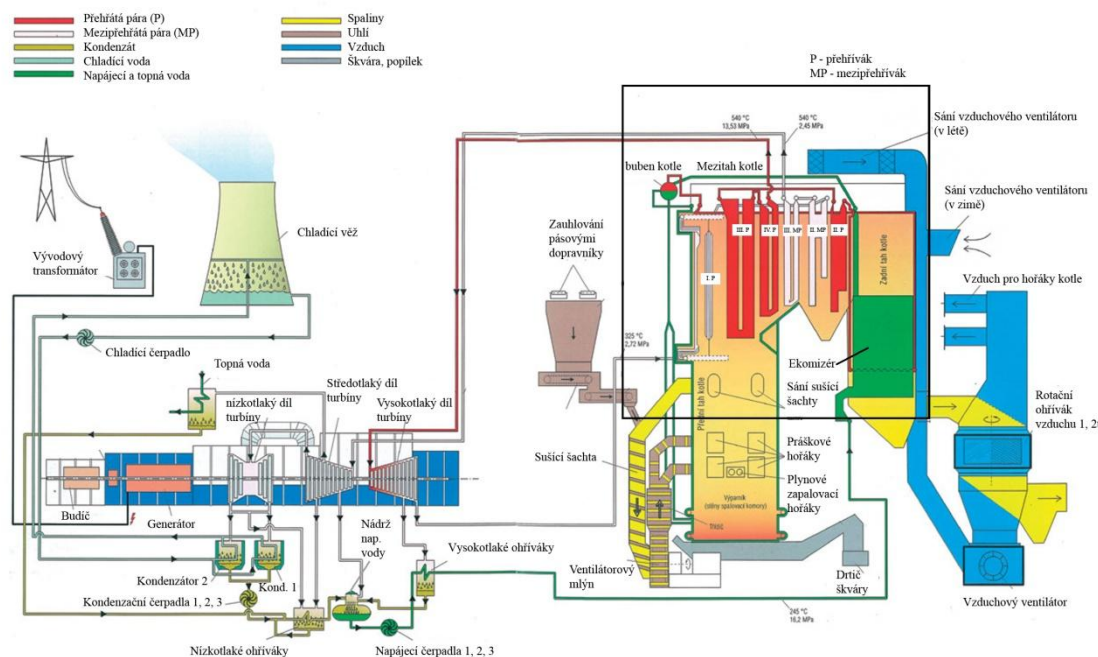


Obr. 3: Typické schéma HIL systému (převzato z [5])

3 Popis modelované soustavy

Modelovanou částí zadané práce je obnovovaná tepelná elektrárna Pruněrov II. Ta se skládá z několika částí, které zde budou popsány.

3.1 Princip tepelné elektrárny



Obr. 4: Zjednodušené schéma výrobního bloku tepelné elektrárny (převzato z [6])

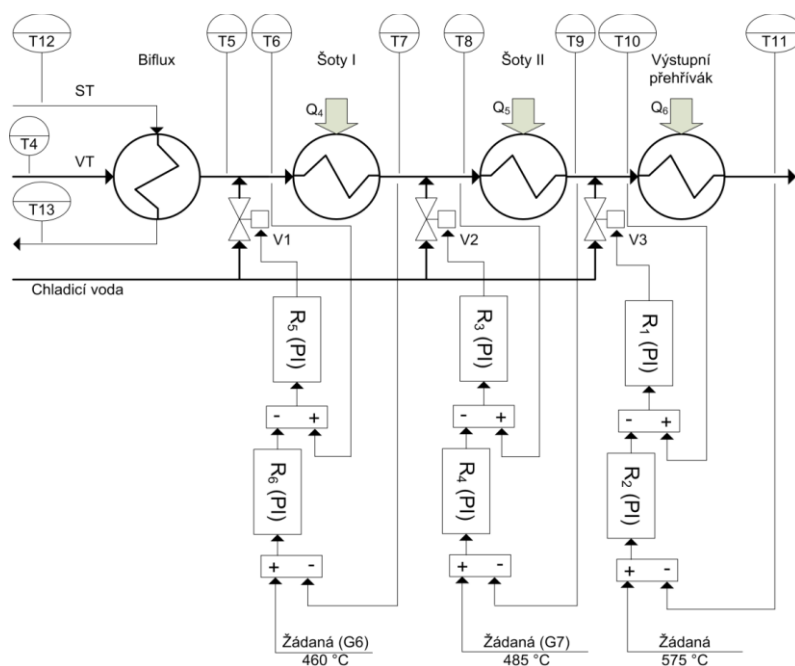
Na obr. 4 je zobrazeno zjednodušené schéma bloku tepelné elektrárny. Pro tuto práci je nejdůležitější částí poslední technologický úsek vysokotlaké (VT) části – šoty I, šoty II a výstupní přehřívák, označené ve schématu (obr. 4) v černém rámečku.

Kotel u tepelné elektrárny může být buď bubnový, nebo průtočný. Bubnový kotel, jak již název napovídá, obsahuje buben, ve kterém dochází ke změně z kapalného skupenství na plynné a dále pokračuje již jen pára. V průtočném kotli dochází ke změně skupenství voda-pára postupně procházením trubkou v částech výparníku a přechodníku. V obou typech kotlů se na začátku používá ekomizér, kde probíhá přehřívání vody. Ta dále vstupuje do výparníku, kde se dle typu kotle mění v páru. Nakonec pára postupuje do přehříváků, kde je pára, pomocí ohřívání spaliny, dodán dostatek energie použitelné k roztáčení turbíny. Teplota této páry je až 550°C. Samostatnou částí je okruh přehřívání páry využívající mezipřehříváky páry. Ty slouží k opětovnému zvýšení teploty páry a následnému zavedení zpět do středotlaké

a nízkotlaké turbíny. Součástí kotle je také protiproudý tepelný výměník typu trubka v trubce, tzv. biflux, zajišťující energetickou výměnu mezi středotlakou (ST) a vysokotlakou (VT) párou. Pára, která již odevzdala svou energii, kondenzuje zpět do kotle. V našem případě je uvažován kotel průtočný [1].

3.2 Model soustavy v Matlabu

Řízený model obsahuje tři výměníky (šoty I, šoty II a výstupní přehřívák) s předřazenými vstříky. Jedná se o poslední technologický úsek VT části. Vytvořením modelu se zabývá disertační práce pana Ing. Lukáše Hubky Ph.D. [1]. Cílem této diplomové práce nebylo vytvořit model, ale připojit řídicí systém již k hotovému modelu. Použitý model je řízen třemi kaskádními PI regulátory, z nichž každou z kaskád lze považovat za samostatný regulační obvod. Do jednotlivých kaskád vstupují konkrétní žádané hodnoty. Zjednodušené schéma je zobrazeno na obr. 5. Pro šoty I je žádaná teplota 460°C, pro šoty II 485°C a pro výstupní přehřívák 575°C. Jako regulované veličiny jsou měřené teploty vždy za konkrétními úseky.



Obr. 5: Blokové schéma regulované VT části (převzato z [1])

4 PLC Siemens S7 – 300

V předchozích kapitolách byl již popsán způsob simulace a simulovaný model. Další důležitou částí této práce je použitý řídicí systém. V tomto případě programovatelný automat od společnosti Siemens.

Programovatelné logické automaty (PLC) jsou řídicí systémy sloužící k řízení technologických a průmyslových procesů. Pro tyto automaty je charakteristické vykonávání programů v cyklech. PLC systémy se většinou skládají z digitálních a analogových vstupů/výstupů. Základní činností PLC je získávání informací ze vstupů (čidel, přepínačů apod.), jejich vyhodnocování a ovládání procesu, nebo zařízení změnou výstupů [7].

4.1 Popis PLC Siemens S7 – 312C

Pro řízení daného modelu bylo použito PLC Siemens řady S7-300. Jedná se o modulární systém, ke kterému je možné připojit další analogové vstupy, výstupy či komunikační moduly. Hlavní částí tohoto PLC je modul obsahující CPU jednotku, která slouží k vykonávání požadovaného programu. Další jednotkou je modul napájení, který napájí PLC 24V. Ke komunikaci s počítačem slouží rozhraní MPI. MPI rozhraní se většinou využívá pouze pro programování. Protože však PC, ke kterému je PLC připojeno, nedisponuje síťovým ani Profibus modulem, tak je v této diplomové práci využito i pro komunikaci. Další rozšiřující moduly nebylo potřeba připojovat, protože při komunikaci přes OPC stačí pracovat pouze s nadefinovanými proměnnými. Samotné PLC je zobrazeno na obrázku níže.



Obr. 6: PLC Siemens S7 - 300

4.2 Programování PLC Siemens S7 - 300

PLC řady S7 – 300 se programují pomocí vývojové platformy Step7. Ta umožňuje programování pomocí několika standardních programovacích jazyků, kterými jsou:

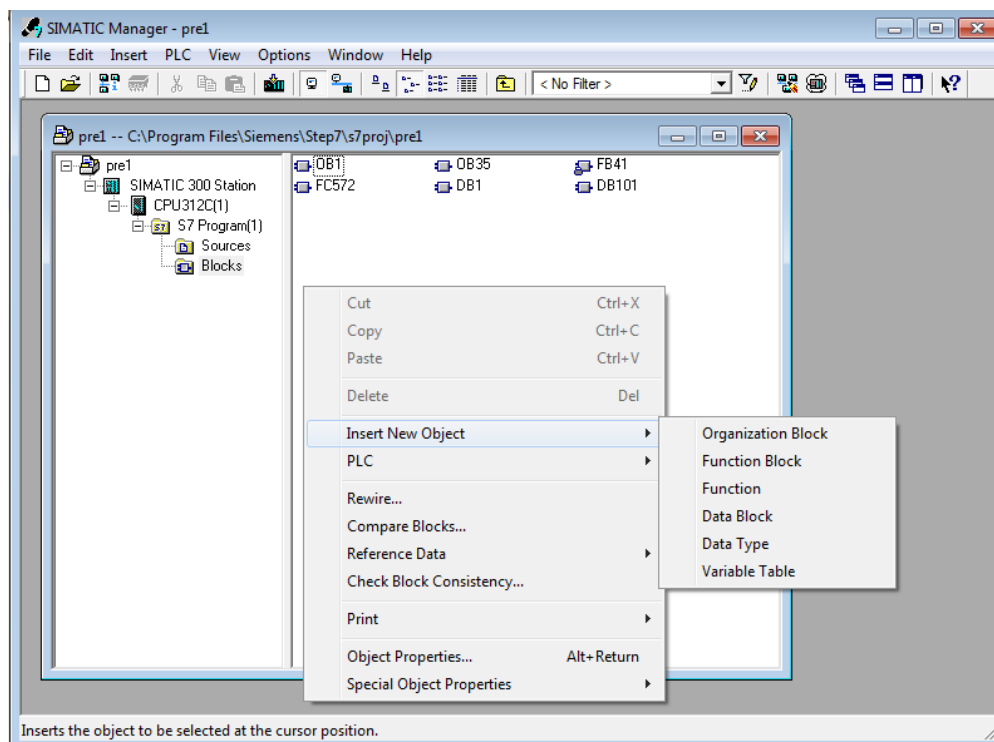
- STL – jazyk podobný Assembleru
- LAD diagram – jazyk podobný reléovým schémátům
- FBD – jazyk pracující s funkčními bloky, podobný LAD

Další možností je využití vyššího programovacího jazyku SCL, který má podobnou strukturu jako Pascal. Není problém v tomto jazyce naprogramovat určitou funkci a tu pak připojit v LAD či STL ke stávající realizaci pomocí vygenerovaného FB, resp. FC. (viz dále). Poslední možností je využití sekvenčního jazyka Graph, který však v této práci využit nebyl. Je vhodný spíše pro čistě logické řízení, kdy je přechod z jednoho bloku do dalšího podmíněn splněním dané podmínky.

Při vytváření projektu lze mezi jednotlivými třemi základními jazyky (STL, LAD a FBD) plynule přepínat. Také není problém jednotlivé části zdrojového kódu napsat v jednotlivých jazycích. Pouze při vytvoření složitějšího kódu v jazyce STL může nastat komplikace při přepínání do FBD, resp. LAD. Ta část kódu potom zůstane zobrazena jako textová struktura STL.

PLC pracuje s klasickými datovými typy, kterými jsou BOOLEAN (BOOL), WORD, DWORD, INT a REAL. Mezi těmito datovými typy lze konvergovat pomocí daných funkcí.

Samotné programování se provádí v tzv. organizačních blocích (OB). Základním blokem obsaženým v každém projektu je OB1. Další druhy bloků se spouští např. při vyvolání přerušení, restartování, chybě PLC či cyklicky dle nastavené doby periody (OB30-OB38). Hlavní okno vývojového prostředí Step7 je zobrazeno na Obr. 7.



Obr. 7: Prostředí Step7

Založený projekt má tvar stromové struktury, kdy je navolen daný typ procesoru a případně další použité moduly (vstupní, výstupní, komunikační apod.). Samotné programování je prováděno ve složkách „Sources“, kde jsou obsaženy zdrojové kódy napsané v jazyce SCL, resp. „Blocks“, obsahující jednotlivé dílčí bloky. Ty lze rozdělit do skupin zobrazených v otevřeném menu. Tedy na Organization Block (OB), Function Block (FB), Function (FC), Data Block (DB) a Variable Table (VAT).

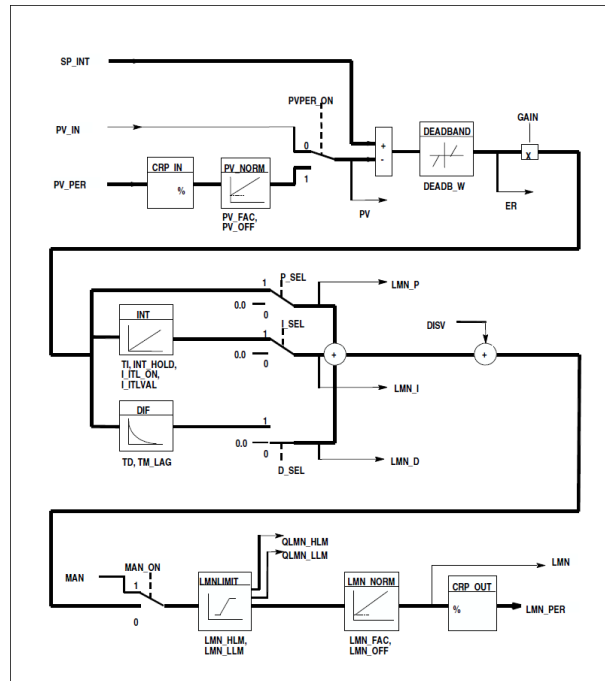
- **OB** – Hlavní část programovací struktury, kde se ze standardních funkcí, jako sčítání, násobení, logické funkce, čítače aj. a dále bloků FC a FB, skládá samotný program. OB bloky jsou vykonávány v cyklech nebo pomocí přerušení.
- **DB** – Slouží k práci s proměnnými a ukládání dat.
- **FB** – Jsou funkční bloky, které se vždy váží na určitý DB, který používají jako paměť pro svou činnost. To jim umožňuje uchovat hodnotu pro více cyklů.
- **FC** – Vykonávají určitou funkci. Na rozdíl od FB nepoužívají DB, ale pracují s dočasnou částí paměti. Při každém spuštění tedy vyžadují vstupní data.
- **SFB, SFC** – Systémové funkce a systémové bloky. Jsou obsaženy přímo v CPU.

4.3 PI(D) regulátor

Při řízení popsané soustavy je využito PID regulátoru obsaženého přímo ve vývojovém prostředí Step7 pod názvem FB41. Tento regulátor se skládá z proporcionální, integrační a derivační složky a zjednodušeně lze jeho vzorec napsat následovně:

$$G_R(s) = K_p \left(1 + \frac{1}{T_i s} + T_D s \right) \quad (1)$$

Blokové schéma tohoto obvodu je zobrazeno na obr. 8. Z tohoto schématu je patrné, že je možné nastavovat velké množství parametrů, jako jsou horní a dolní limity integrační složky, připojit poruchovou veličinu, přepínat na manuální režim apod. Kompletní výčet všech parametrů je zobrazen v přehledné tabulce v příloze D. Z této tabulky také např. vyplývá, že derivační složce je možné pomocí parametru TM_LAG přidat navíc derivační zpoždění.



Obr. 8: Blokové schéma regulačního bloku FB41 (převzato z [7])

4.4 Základní pojmy používané při regulaci

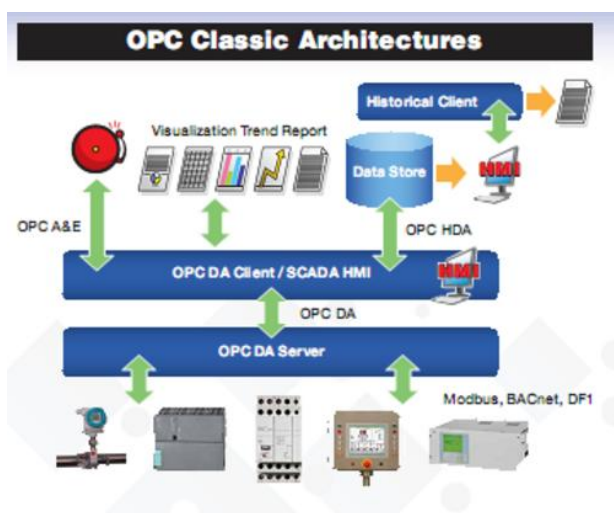
Při řízení se snažíme vnějším působením na danou soustavu dosáhnout požadovaného stavu. Pro regulaci se používá standardní označení jednotlivých částí regulované soustavy, které bude použito i v této práci:

- **Regulovaná veličina (y)** – vstupní hodnota regulátoru, výstup z řízené soustavy (Process Variable – PV_IN)
- **Řídicí veličina (w)** – žádaná hodnota (Set Point – SP_INT)
- **Regulační odchylka (e)** – rozdíl mezi žádanou a regulovanou veličinou (Error Signal - ER)
- **Akční veličina (u)** – výstupní hodnota regulátoru, vstup řízené soustavy (Manipulated Value - LMN)
- **Poruchová veličina (d)** – může působit v libovolném místě reg. soustavy (Disturbance Variable - DISV)

5 OPC technologie

OPC je standard průmyslové komunikace, který vznikl ve spolupráci společnosti Microsoft a mnoha světových dodavatelů automatizačních prostředků. Jeho hlavním přínosem je snaha o komunikaci mezi jednotlivými hardwarovými a softwarovými prostředky bez závislosti na konkrétním výrobcí. Tím pádem odpadá potřeba, aby každá aplikace obsahovala ovladač pro konkrétní zařízení. Díky tomu je velmi výhodné jeho využití pro řízení a sledování technologických procesů [2].

OPC využívá klasickou architekturu klient/server. OPC server je software, který slouží jako prostředník při komunikaci mezi průmyslovým zařízením a klientem v PC. K jednomu serveru se může připojit více klientů od různých výrobců. Stejně tak se jeden klient může připojit k více serverům. Klientem mohou být různé vizualizační SCADA programy nebo např. i Matlab s nainstalovanou nadstavbou „OPC Tools“, který bude použit v této práci. Pomocí OPC rozhraní můžeme proměnné z PLC v počítači jak sledovat, tak i měnit. OPC architektura je dobře patrná na obr. 9.

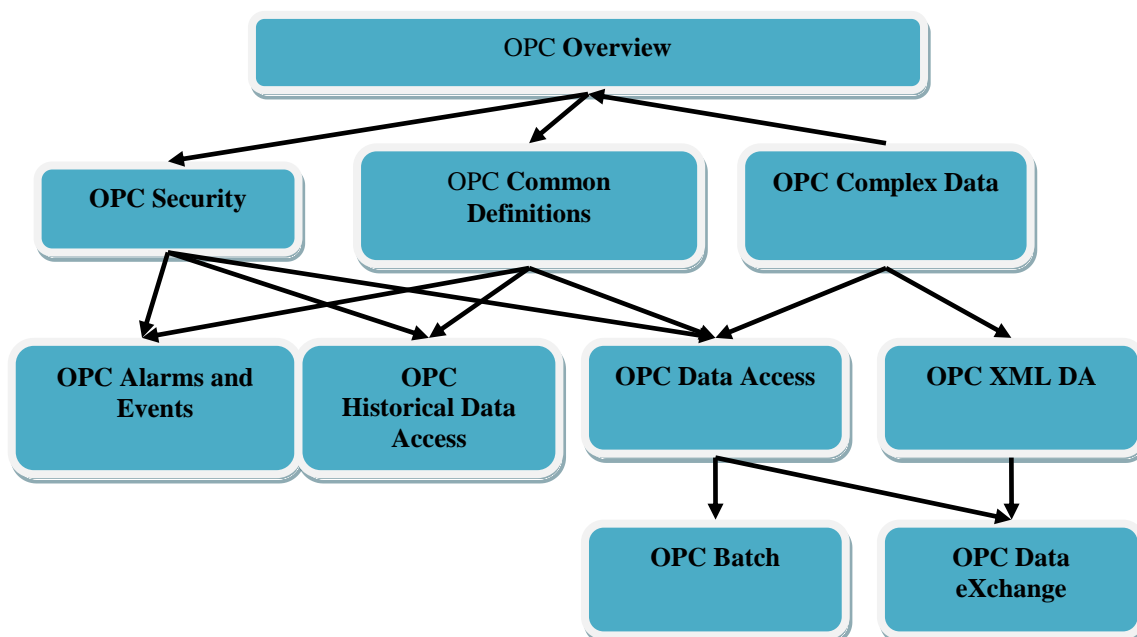


Obr. 9: Architektura OPC (převzato z OPC Foudation [9])

5.1 Specifikace OPC

Standard OPC je vytvářen a udržován pomocí specifikací OPC, o které se stará mezinárodní dobrovolná organizace OPC Foundation [9]. Protože standard OPC vznikl ve spolupráci se společností Microsoft, je tedy založen na jejich metodách OLE, COM a DCOM. COM je metoda vývoje software složeného z malých spustitelných částí zdrojového kódu, který poskytuje služby ostatním komponentám – svým klientům.

Specifikace OPC dobře zobrazuje následující obrázek (obr. 10).



Obr. 10: Specifikace OPC a vazby mezi nimi

Základem specifikace je OPC Overview obsahující souhrn informací o výhodách, účelu a vlastnostech OPC. Další body specifikace pak jsou:

- *OPC Security* – specifikace zabezpečení OPC serveru
- *OPC Historical Data Access* – specifikace určující nakládání s uloženými daty v určité databázi
- *OPC Complex Data* – společná specifikace pro OPC DA a XML DA sloužící serveru k práci se složitějšími datovými strukturami jako jsou např. XML dokumenty
- *OPC Data Access* – specifikace sloužící k přenosu dat mezi PLC a dalšími klienty v reálném čase
- *OPC Alarms&Events* – specifikace potvrzení událostí a alarmů
- *OPC Batch* – dodatek specifikace Data Access pro dávkově řízené procesy (používáno v potravinářství či farmacii)
- *OPC Data eXchange* – specifikace komunikace mezi více servery přes Ethernet
- *OPC XML DA* – specifikace obsahující pravidla a formáty pro práci s XML

Nejnovější generace OPC standardu se nazývá „OPC Unified Architecture“. Tato architektura již není závislá na metodách COM a DCOM, nýbrž je založena na webových službách a využívá jazyka XML.

5.2 Popis specifikace OPC Data Access

Tato specifikace, jak již bylo řečeno výše, slouží k přístupu k datům v reálném čase. Je nejčastěji používanou specifikací a také jednou z prvních, které vznikaly. Definuje rozhraní mezi klientem a servery, kteří poskytují procesní data.

Mezi DA serverem a DA klientem se uplatňují tři druhy datové výměny. Konkrétně se jedná o synchronní čtení/zápis, asynchronní čtení/zápis a obnovení dat. Čtení těchto hodnot je možné provádět přímo z datového zdroje (PLC) nebo z interní cache paměti serveru pro zpracování dat. Zápis je prováděn vždy přímo do datového zdroje.

Při synchronním a asynchronním přenosu se klient přizpůsobuje chování zdroje dat. V případě synchronní komunikace klient zavolá metodu a očekává vrácení hodnoty. Synchronní volání je vhodné používat při rychlém přístupu k datům, jinak je klient na velkou dobu zablokován. Proti tomu při asynchronním režimu klient volá metodu a okamžitě dostane odpověď. Odpovědí není však požadovaná hodnota, pouze potvrzení o možnosti čtení či zápisu hodnoty. Hodnota je odeslána v čase určeném nastavenou hodnotou přenosu. Asynchronní režim je tím pádem vhodnější, pokud jsou data dostupná v delších intervalech [9].

5.3 Výběr OPC serveru

Pro tuto práci je důležité vybrat vhodný OPC server. Na trhu se jich vyskytuje velké množství od různých výrobců a tyto servery se od sebe samozřejmě dost odlišují. Hlavní podmínkou je bezproblémový provoz s PLC od firmy Siemens.

Většina výrobců nabízí zkušební verzi, běžící omezenou dobu od spuštění, zdarma. Např. Deltalogic (90 minut), KepServerEX OPC (120 minut), Softing – S7/S5 OPC Server (90 minut), KOS EC1 (120 minut), VIPA OPC Server (24 hodin). Společnost Matrikon nabízí 30 denní trial verzi. Tu je možné na požádání získat i u serverů Deltalogic a KepServerEX OPC. Pouze Siemens u svých serverů takovou možnost nenabízí. Siemens totiž neprodává OPC server, ale komunikační rozhraní,

jehož součástí je OPC server. Dalším důležitým parametrem při výběru bylo, aby server nabízel možnost jednoduchého načtení proměnných přímo ze zdrojového kódu projektu Step7. V neposlední řadě bylo potřeba, aby server dokázal komunikovat i přes rozhraní MPI, což u všech serverů není samozřejmostí.

Po přečtení několika recenzí a samotném vyzkoušení se jevil jako nejrozumnější server Deltalogic, který nabízí bezproblémové nastavení a zprovoznění serveru a zároveň dokáže načíst proměnné přímo z projektu Step7. I možnost komunikace po MPI je bezproblémová [10].

5.4 OPC server Deltalogic

Jedná se o OPC server určený ke komunikaci s PLC Siemens Simatic S5/S7 zajišťující datový přenos mezi OPC klientem a PLC Siemens S7-200 (300, 400). Přes jeden server je možno připojit najednou až 256 PLC. Tento server nabízí celou řadu možností jak se připojit k danému PLC Siemens. Např. přes převodník MPI, Profibus, nebo Ethernet. Bližší je tento server a jeho nastaven popsán při řešení komunikace s tímto serverem.

6 Nastavení komunikace mezi OPC serverem a klienty

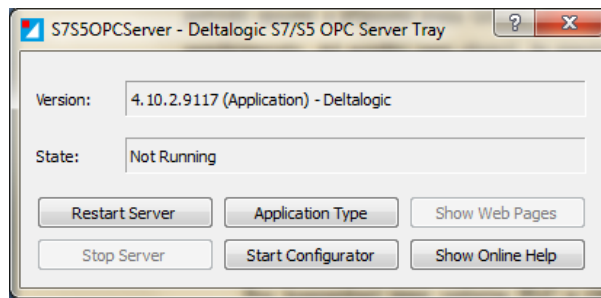
Základním předpokladem pro realizaci diplomové práce je zprovoznění komunikace. Jak již bylo několikrát zmiňováno o komunikaci se stará OPC server Deltalogic, ke kterému jsou připojeny na jedné straně Matlab a na druhé PLC Siemens.

6.1 Propojení OPC – Siemens

PLC Siemens Simatic S7-300 může být k počítači připojeno několika způsoby. Nejčastějším způsobem je připojení přes rozhraní MPI (pro programování). Dalšími možnostmi je využití rozhraní Profibus nebo Ethernet. V případě uvažovaném v této práci bylo PLC připojeno přes rozhraní MPI.

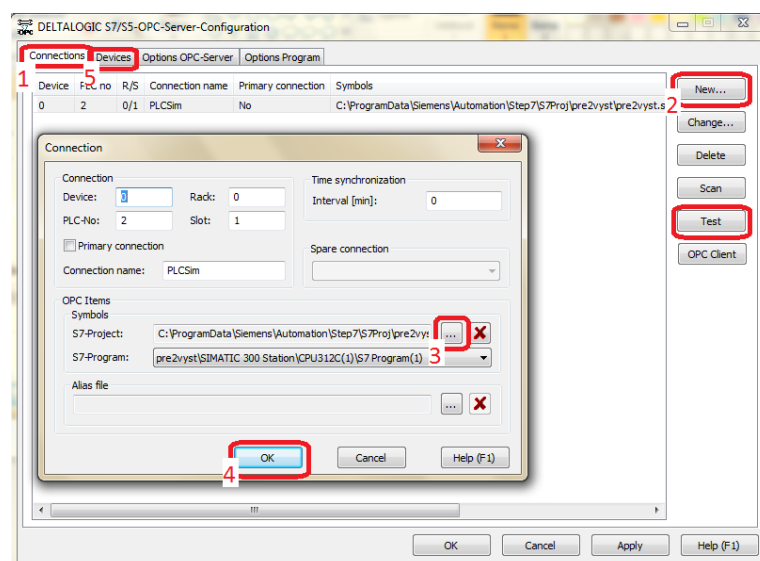
Jako OPC server je možno využít některý z velké nabídky serverů od různých výrobců. Vlastní OPC server má například i společnost Siemens nabízený pod názvem WinCC. V práci použitý OPC server je však od společnosti Deltalogic. Ta nabízí zkušební verzi, která běží 1,5 hod a poté je potřeba ji restartovat. Na počítači, na kterém jsem pracoval, byla však nainstalovaná plná verze. Na mém osobním počítači jsem zkoušel tuto zkušební verzi v kombinaci se simulovaným PLC „S7-PLCSIM“, které je součástí balíčku Step7. Zde se vyskytl zajímavý problém. Při nastavení komunikace se toto simulované PLC jevílo serveru Deltalogic jako připojené. Při testu komunikace server potvrdil spojení a připojení psalo GOOD. Přesto se však posílané hodnoty na serveru nezobrazovaly. Až později jsem objevil, že simulované PLC nepodporuje komunikaci přes OPC, ačkoliv se tvářilo, že ano. Zjistil jsem, že alternativní možností je použití komerčního simulátoru PLC s názvem „ACCONtrol“, taktéž od společnosti Deltalogic. Bohužel jeho bezplatná zkušební verze funguje nepřetržitě cca 15 minut a potom je potřeba restart aplikace. Z tohoto důvodu se jednalo téměř o nepoužitelné řešení, které jsem využíval pouze v případě, kdy se vyskytly problémy se vzdálenou plochou u pracovního počítače fyzicky přítomného ve škole.

Pro komunikaci mezi reálným PLC a OPC serverem Deltalogic je postup následovný. Při spuštění OPC serveru se na liště Windows objeví modrá ikona s červeným čtverečkem signalizující, že je server v neběžícím režimu. Po dvojitým kliknutí vyskočí následující okno (obr. 11).



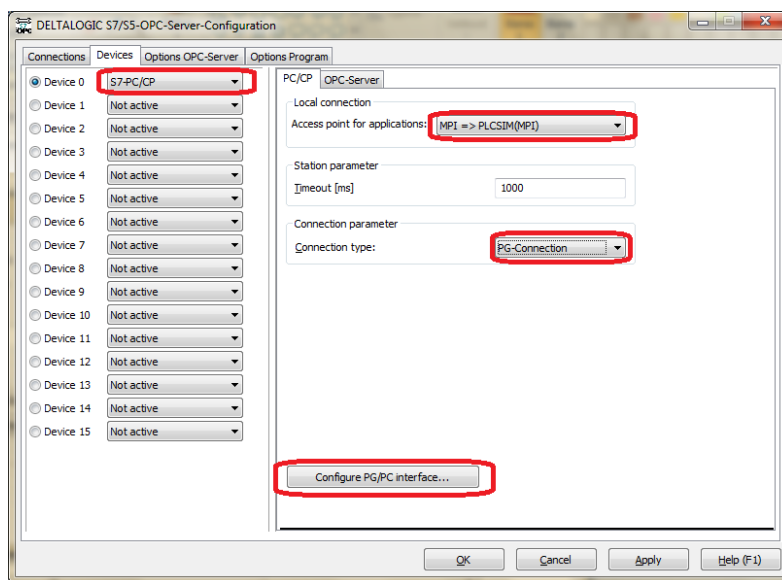
Obr. 11: Startovní okno OPC serveru Deltalogic

Pomocí tlačítka „Restart Server“ se provede spuštění serveru. Předtím je vhodné zkontrolovat jeho nastavení přes tlačítko „Start Configurator“, případně při prvním spuštění ho poprvé nastavit (obr. 12). Po kliknutí na toto tlačítko bychom se měli nacházet v záložce „Connection“ (1). Pokud máme již nějaký server nakonfigurovaný, bude v seznamu v této záložce. Pokud ne, vytvoříme nový pomocí tlačítka „New“ (2). Otevře se okno s názvem „Connection“. Zde vyplníme parametry jako číslo „Device“ v případě používání více zařízení (číslo zařízení zvolené následně v záložce „Devices“), „Rack“ při použití více slotů v modulárním PLC, „PLC-No“ a „Slot“ což jsou identifikační čísla pro zařízení S7. Následně je potřeba vybrat cestu ke Step7 souboru našeho projektu (3). Toto je velmi důležité pro identifikaci použitých proměnných v dalších připojených zařízeních. Vše potvrdíme tlačítkem „Ok“ (4). Na záložce „Connection“ je ještě zajímavé tlačítko „Test“ sloužící pro ověření spojení s PLC. Při prvním spuštění by kliknutí na něj mělo zahlásit chybu, protože jsme ještě neprovedli konfiguraci připojení, která se provádí v záložce „Devices“ (5).



Obr. 12: Konfigurace OPC serveru - vytvoření nového připojení

Po kliknutí na záložku „Devices“ se dostaneme na stránku, kde nastavujeme pro jednotlivá zařízení způsob, jakým se serverem komunikují. V případě komunikace přes MPI je vhodné zvolit možnost „S7 PC/CP“. Zde potom nastavíme „Access point for applications:“ na „MPI => PC Adapter(MPI)“ a překlikneme „Connection type“ na „PG-Connection“. V případě problému např. s přístupovým bodem nebo s komunikací můžeme na této záložce kliknout na „Configure PG/PC Interface“ a pokusit se nastavit požadované připojení.



Obr. 13: Konfigurace OPC serveru - konfigurace zařízení

Když se teď vrátíme na záložku „Connections“, tak už by při kliknutí na tlačítko „Test“ mělo vyskakující okno zahlásit, že je vše v pořádku. Po restartování serveru by již neměly nastat komplikace s komunikací. Za zmínku ještě stojí možnost sledovat online hodnoty posílané na server přes tlačítko „OPC Client“ obsažené opět v „Start Configurator->Connections->OPC Client“. Sledování komunikace je také možné přes webové rozhraní kliknutím v hlavním okně programu na „Show Web Pages“. Zde je potřeba se přihlásit. Standardně nastavené přihlašovací údaje jsou:

User: **Operator**

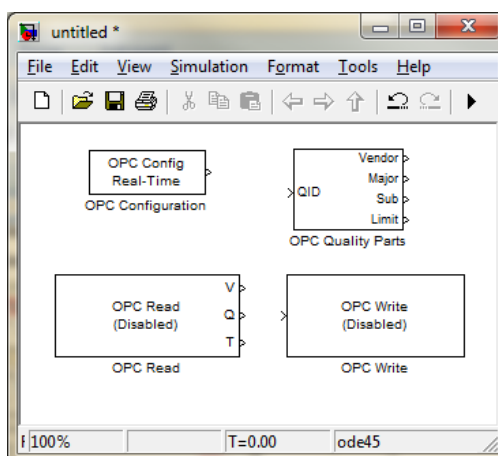
Password: **op**

Následně již není problém sledovat požadované proměnné.

6.2 Propojení OPC server – Matlab

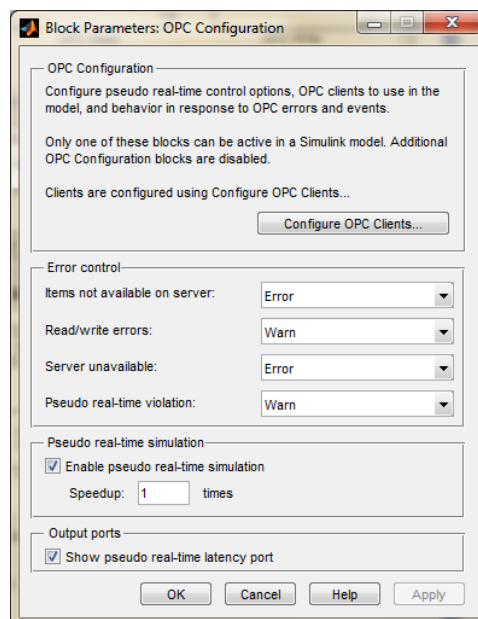
Po nastavení serveru je potřeba zprovoznit komunikaci ze strany Matlabu. Jednou z možností je nainstalovat potřebný „OPC Toolbox“. Případně existují i alternativy, jako je „IPCOS OPC for Matlab“, který však nebyl v této práci testován a tedy ani použit.

Samotný OPC Toolbox obsahuje několik komponent – bloků. Jednotlivé komponenty se nazývají „OPC Read, OPC Write, OPC Config a OPC Quality Strings“. Další součástí Toolboxu je několik funkcí sloužících pro komunikaci z příkazové řádky. Jak již názvy napovídají, slouží ke čtení, resp. zápisu dat na server, k nastavení připojení a zobrazení kvality přenosu. Za zmínku stojí možnost vyvést z bloku OPC Config hodnotu na displej, která udává, jestli není problém s vykonáváním v pseudo-reálném čase. Pokud je hodnota na výstupu tohoto bloku záporná, tak se proces vykonává delší dobu, než je reálný čas a simulace neprobíhá korektně [11].



Obr. 14: Bloky OPC Toolbox

Prvním blokem tohoto Toolboxu je „OPC Config“. Po jeho otevření vyskočí následující okno (obr. 15).



Obr. 15: Nastavení parametrů OPC Toolboxu

Ke konfiguraci slouží tlačítko „Configure OPC Clients“. Po kliknutí na toto tlačítko vidíme všechny servery, které jsou již nakonfigurovány, případně pomocí tlačítka „Add“ můžeme přidat další. Pokud je vše v pořádku, v seznamu se objeví běžící server Deltalogic a stačí ho pouze vybrat. Po potvrzení a návratu na původní okno (obr. 15) můžeme nastavit další parametry Toolboxu. „Error control“ udává, jaká akce má být provedena v případě nějakého problému (akcí je rozuměno, jestli Matlab zahlásí chybu, varování nebo nic neprovede). Těmito „problémy“ jsou např.: nenalezení požadované proměnné, problém se čtením/zápisem či s připojením na server nebo nestíhání vykonávání simulace v pseudo-reálném čase.

Blok „OPC Read“, jak název napovídá, slouží ke čtení hodnot ze serveru. Zde je možnost nastavit jednotlivé možnosti čtení, které byly uvedeny již v kapitole pojednávající o OPC serverech obecně. Konkrétně se jedná o synchronní/asynchronní přenos s možností čtení přímo ze zařízení či interní cache.

Posledním důležitým blokem je „OPC Write“, který slouží k zápisu hodnoty na server a umožňuje opět pracovat v synchronním nebo asynchronním režimu.

6.3 Ověření kompletní komunikace

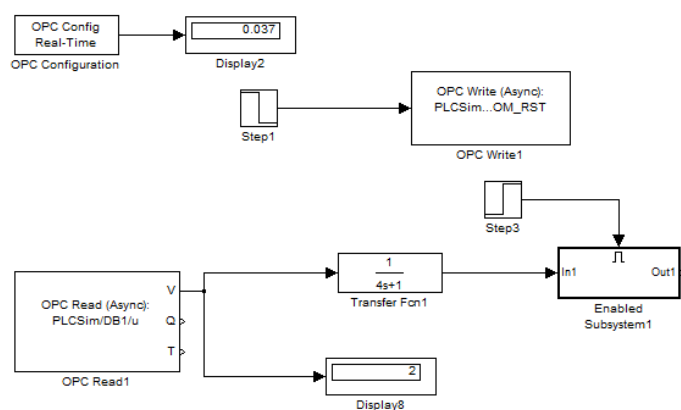
Nejprve byl vytvořen v prostředí Step7 nový projekt, který obsahuje objekt OB35 sloužící k vykonávání procesů v cyklech dle nastavené hodnoty. V tomto případě

100ms. Pomocí tabulky DB1 jsou nadefinované jednotlivé vstupy a výstupy jako logické proměnné. V objektu OB35 je dále zapojena struktura s jedním XOR hradlem s přivedenými konkrétními proměnnými. Do stávajícího projektu byl ještě v bloku OB35 přidán funkční blok FB41 sloužící jako PI regulátor s řadou parametrů, které je možné nastavit (Příloha D). Ke každému vloženému PID bloku musí být přiřazen určitý DB (data blok). Ten se sám vygeneruje při vložení FB do projektu. Zde je potřeba dbát na správné nastavení všech potřebných parametrů. Některé hodnoty jsou ve výchozí poloze zapnuty, jiné vypnuty. Důležité je si ohlídat hlavně parametr MAN_ON, který je standardně nastaven na TRUE. Základní nastavení daného PI regulátoru je zobrazeno níže. Tento regulátor slouží pouze k ověření komunikace a slouží k řízení přenosu:

$$F(s) = \frac{1}{4s+1} \quad (2)$$

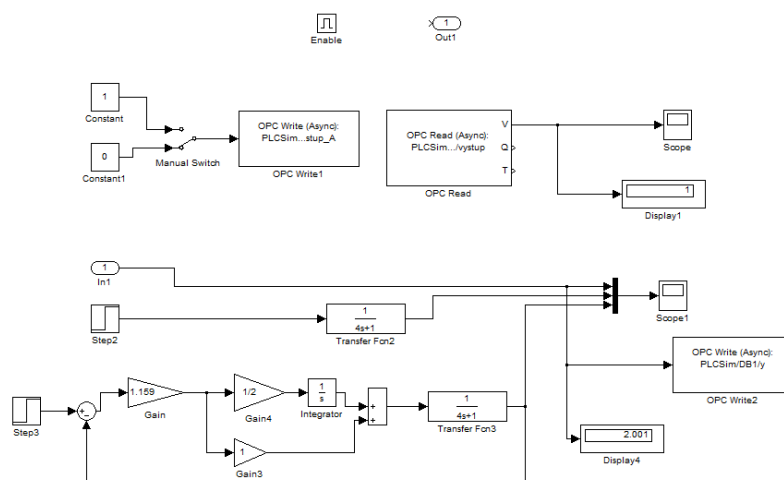
```
CALL  "CONT_C", DB101
      COM_RST   :=
      MAN_ON    :=FALSE
      P_SEL     :=TRUE
      I_SEL     :=TRUE
      D_SEL     :=FALSE
      CYCLE     :=T#100MS
      SP_INT    :=2.000000e+000
      PV_IN     :=DB1.DBD2
      GAIN      :=1.159000e+000
      TI        :=T#2S
      LMN       :=DB1.DBD6
```

V Matlabu byl vytvořen také nový projekt. Ten slouží k simulaci dané soustavy a ověření komunikace. Na prvním obrázku (obr. 16) je zobrazena hlavní část projektu, kdy je PI regulátor nejprve resetován a následně dochází ke spuštění celé simulace obsažené v bloku „Enabled Subsystem 1“. Struktury v tomto bloku jsou vykonány až po zvoleném čase udaném v bloku Step.



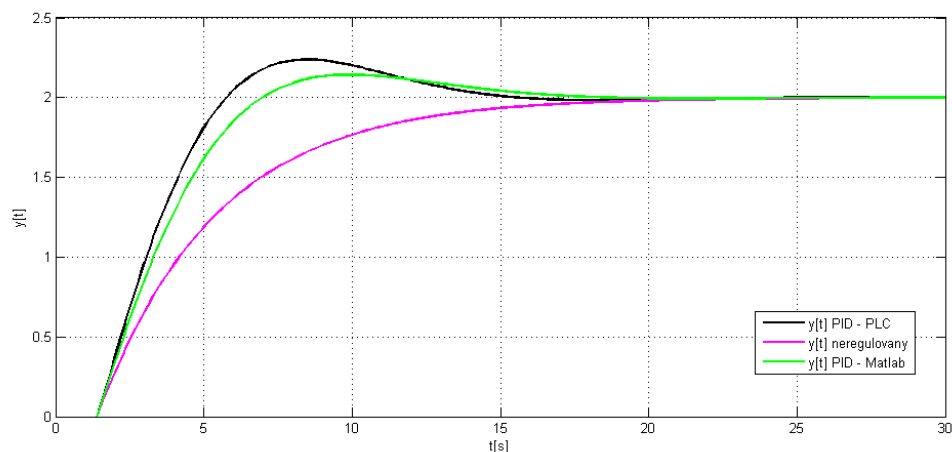
Obr. 16: Ověření komunikace Matlab - PLC přes OPC

V bloku „Enabled Subsystem 1“ (obr. 17) je přidán přepínač, kterým se posílá logická hodnota na jeden ze vstupů hradla XOR v PLC. Potom je zde také zapojena neregulovaná soustava, soustava regulovaná pomocí PLC a soustava regulovaná pomocí ideálního PI regulátoru z bloků v Matlabu. Pro porovnání průběhů jsou všechny tři průběhy zapojeny do Scope 1.



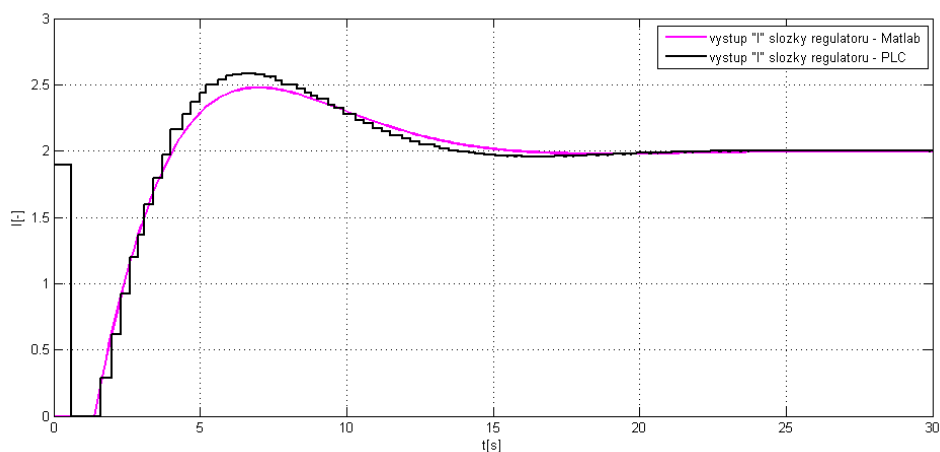
Obr. 17: Ověření komunikace Matlab - PLC přes OPC (Enabled Subsystem)

Výsledná odezva na připojený skok je zobrazena na následujícím grafu (obr. 18). Černá odezva je řízená v PLC, zelená v Matlabu. Růžová je neregulovaná soustava. Posunutí z počátku je z důvodu prodlevy při resetování regulátoru v PLC.

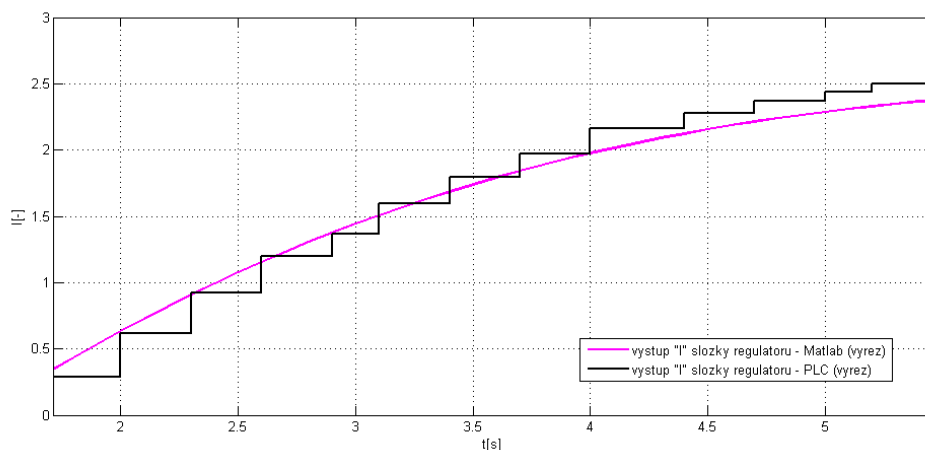


Obr. 18: Přechodové charakteristiky úlohy k ověření komunikace

Na níže znázorněném grafu je zobrazeno, jakým způsobem se mění výstupy z „I“ složky regulátorů v Matlabu a v PLC. Vzorkování v OPC Toolbox je nastaveno na 0,1 s. Jak je však z grafů patrné (obr. 19, resp. obr. 20), tak vzorky mají vyšší periodu vzorkování. V příkazové řádce Matlabu také vyskočila řada varování oznamujících, že simulace neprobíhá v určitých časových skocích zcela korektně (*Warning: block 'komunikace1/OPC Configuration': Pseudo real-time violation at simulation time 29.90*).

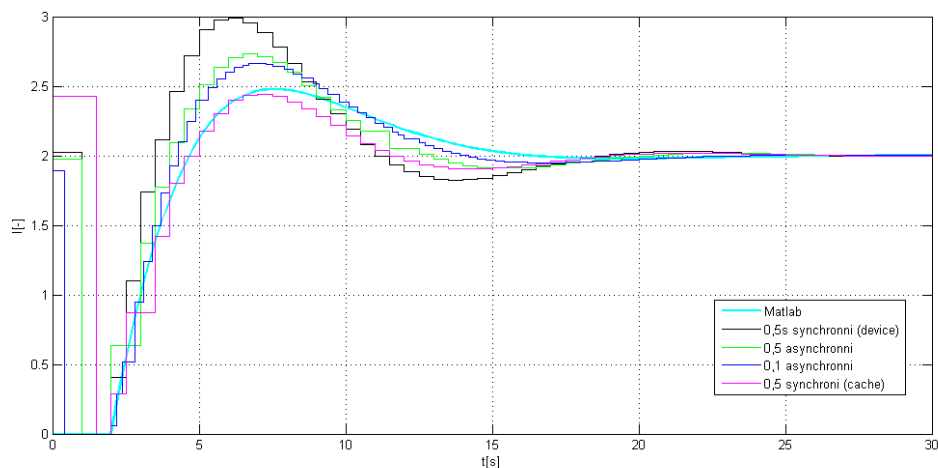


Obr. 19: Porovnání výstupů „I“ složek regulátorů Matlab – PLC



Obr. 20: Výřez průběhů „I“ složek regulátorů Matlab – PLC (vzorkování 0,1s)

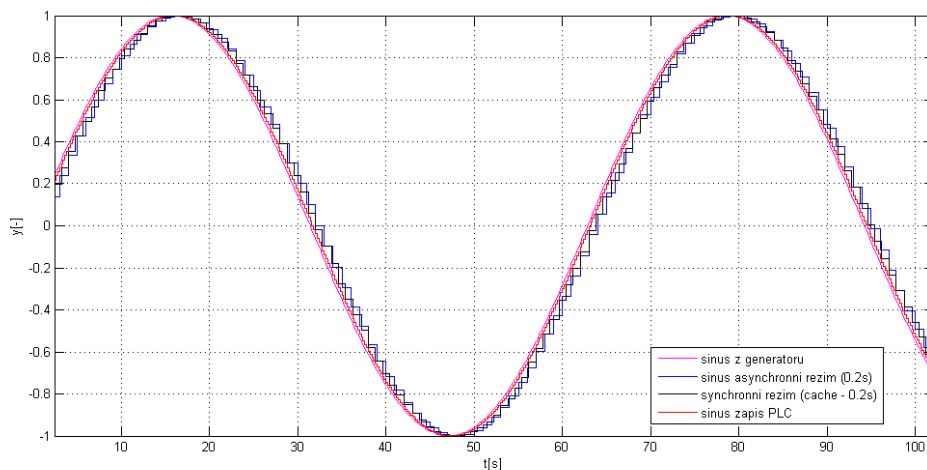
Proto jsem změnil periodu vzorkování na 0,5 s a vyzkoušel různé možnosti čtení proměnných ze serveru. Výsledky jsou shrnuty na obr. 21. Z těchto testů vyplynulo, že synchronní režim ve verzi „device“ vůbec nestíhá vykonávat v daném pseudo-reálném čase (na výstupu bloku „OPC config“ naskočí vysoká záporná hodnota signalizující nestíhání). V ostatních režimech je vykonávání v pseudo-reálném čase dle výstupu „OPC Config“ celkem korektní, avšak výstupy z regulátoru se ideálnímu pouze přibližují.



Obr. 21: Porovnání různých výstupů "I" složky

Ačkoli na stránkách Matlabu uvádějí, že vykonávat simulace v čase, např. 0,2 s v režimu „synchronous (device)“, by neměl být problém [12], což však při testech se zařízením použitým v této práci nemohu potvrdit. Provedl jsem ten samý test, jako ve jmenovaném článku. Připojil jsem generátor sinusového signálu, který jsem posílal přes

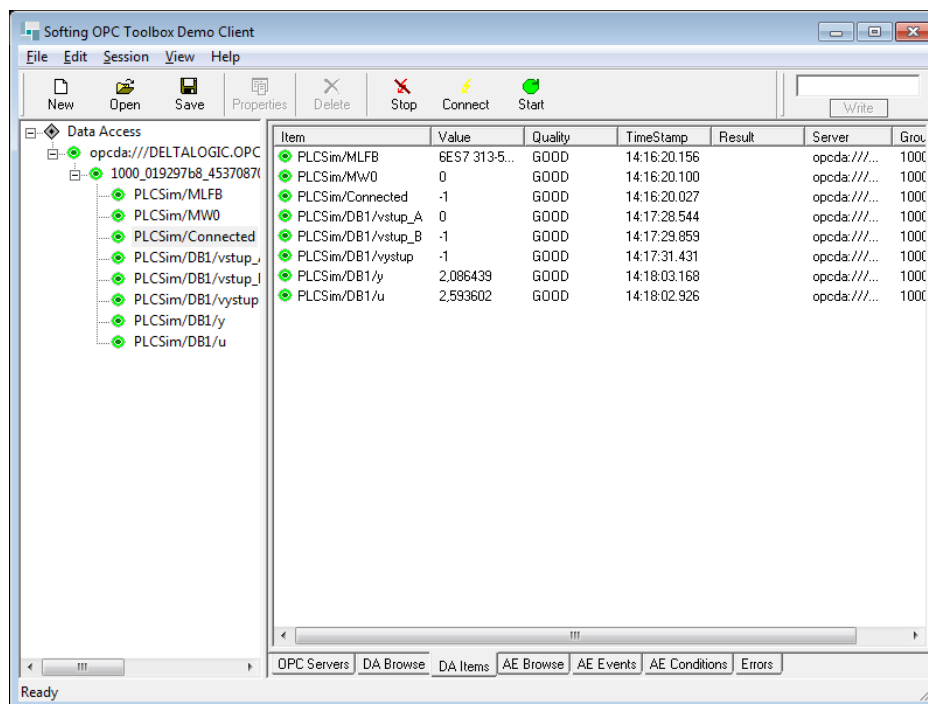
OPC server do PLC. V PLC bylo nastaveno vynásobení příchozí proměnné jedničkou a uložení do druhé proměnné. Na níže znázorněném grafu (obr. 22) je růžově zobrazen průběh proměnné posílané do PLC (a následně čtené) a černě (resp. modře) výstup z druhé proměnné, který je pouze vynásoben jedničkou. Proces v PLC se vykonává v cyklech 200 ms. Při testování s nastavením „synchronous (device)“ byl výstup z „OPC Config“ opět na velmi vysoké záporné hodnotě signalizující velké nestíhání vykonávání v pseudo-reálném čase. V režimu „synchronous (cache)“ nestíhání není signalizováno, ale průběh vychází téměř totožný. Výstupní hodnota z PLC je vzorkována přibližně 1 s. Z těchto výsledků usuzuji, že je problém spíše ve vykonávání v PLC, než v serveru, resp. Matlabu.



Obr. 22: Simulace sinusového signálu

Konkrétně problémem s korektním vykonáváním simulace se zabývá následující článek [13]. Je zde uvedeno, že pokud to simulaci nevadí, tak je vhodné nastavit vyšší periodu vzorkování a spouštět simulaci v asynchronním režimu. Z výše popsanych testů vyplývá, že mezi asynchronním režimem a synchronním „cache“ není zas tak velký rozdíl.

Všechny hodnoty je možné také sledovat pomocí Toolboxu obsaženého v OPC serveru Deltalogic (obr. 23). Zde je také možno sledovat, na jaké hladině se nachází logické proměnné.



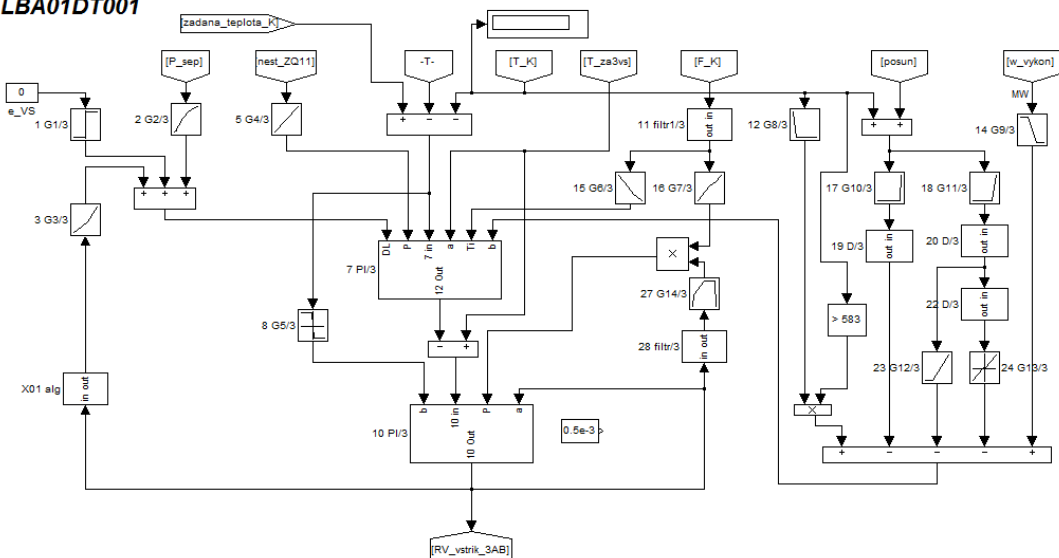
Obr. 23: Softing OPC Toolbox Demo Client - sledování proměnných na OPC serveru

7 Sestrojení regulátoru v PLC

V dodaném Matlab modelu je obsažen již vyprojektovaný řídicí systém (viz Příloha B). Tento řídicí systém je takové struktury, která se standardně používá při řízení obdobných zařízení. Hlavním úkolem a cílem této práce je dle dodané projektové dokumentace vytvořit řídicí systém také v PLC Siemens. Při testování komunikování přes OPC neprobíhala komunikace vždy úplně korektně. Po odzkoušení různých nastavení komunikace vyšlo jako nejlepší nastavení v OPC Toolbox na režim „Asynchronous“ s periodou vzorkování 1 sekunda, přičemž program v PLC běžel ve smyčce 100 ms. Při tomto nastavení byly vzorky opravdu velké 1s, a protože pro tuto práci jsou zajímavé výsledky v delších časových úsecích, tak to nebylo zase tolik omezující.

Jedná se o kaskádní zapojení třikrát dvou PI regulátorů, které pomocí snímačů v jednotlivých úsecích šoty I, šoty II a výstupního přehříváku řídí ventily vstřikující do jednotlivých částí chladicí vody.

Pro simulaci a vyzkoušení řídicích algoritmů byla nakonec vybrána část výstupního přehříváku. Ta má ze všech tří částí nejsložitější strukturu a z její realizace není problém případně sestavit řízení šoty I a šoty II. Řídicí obvod se skládá z několika druhů bloků, které je potřeba postupně sestavit v daném PLC a otestovat. Schéma výstupního přehříváku je zobrazeno na obr. 24, z něhož je zřejmé, že neobsahuje pouze PI regulátory zapojené do kaskády, ale i další bloky, ze kterých patří většina mezi funkční generátory. Dalšími bloky jsou sčítací a násobící členy, filtry a derivační členy.



Obr. 24: Řídicí obvod výstupního přehříváku

7.1 Popis řídicího obvodu výstupního přehříváku

V této kapitole je popsáno schéma řídicí části výstupního přehříváku (obr. 24) dle dodané technické dokumentace.

Žádaná hodnota teploty páry za výstupním přehřívákem je proměnlivá a je generována ve speciálním obvodu. K regulační odchylce, počítané z této žádané hodnoty a teploty páry za kotlem, se připočítává dynamická urychlující dopředná vazba, a to za účelem zrychlení odezev elektrického výkonu bloku. Zmíněnou regulační odchylku zpracovává regulátor vnější smyčky (blok 7). Regulátor vnější smyčky má navíc proměnlivou dolní hodnotu limitu výstupu regulátoru. Dolní limit je odvozen jako součet tří složek vstupů. První složkou je generátor G1, který kompenzuje parazitní vliv vstříkové regulace sousední větvě. Druhá složka, generátor G2, se odvozuje od tlaku páry za separátory tak, aby žádaná hodnota pro regulátor malé smyčky byla v oblasti mírně přehřáté páry. Třetí složka dolního limitu se odvozuje v G3 od výstupu regulátoru vnitřní smyčky, tj. od žádané polohy vstříkového ventilu. Horní limit regulátoru je konstantní. Dále do tohoto regulátoru vstupuje urychlující dopředná vazba (sumátor 25) složená z pěti složek, které reagují na změnu žádané hodnoty elektrického výkonu (G9) a na teplotu páry za přehřívákem (G10, G11, G12, G13). Generátor G10 propouští teploty v úzkém pásmu nad nominální teplotou a ty se derivují v derivátoru (blok 19).

Generátor G11 propouští teploty v širším pásmu kolem nominálních hodnot a ty se derivují v derivátoru (blok 20). V generátoru G12 se vybírají opravy od provozně rizikových trendů teploty. V derivátoru (blok 22) se provádí druhá derivace teploty za přehřívákem. Za tímto derivátorem je zapojen generátor G13, kde se necitlivostí generátoru potlačují opravy od malé fluktuace druhé derivace teploty. Na sumátor 25 se tedy propustí jen provozně významné změny, které je potřeba vazbou utlumit. Zesílení regulátoru vnější smyčky je odvozeno od signálu „nestabilita“ tvořeného v organizátoru regulací. Za neklidného provozu bloku se zesílení snižuje. Integrační časová konstanta se odvozuje v generátoru G6 v závislosti na průtoku páry kotlem.

Regulátor vnitřní smyčky (blok 10) pracuje podle regulační odchylky teploty páry za vstřikem. Dopřednou vazbou tohoto regulátoru je v generátoru G5 upravená regulační odchylka vnější smyčky, která slouží jako náhrada derivační složky. Zesílení vnitřní smyčky se odvozuje od maxima polohy vstřikových ventilů a průtoku páry kotlem. Integrační složka je konstantní.

7.2 Testování dílčích částí realizace – funkční generátory

Generátory jsou ve schématu označeny G1 – G14, které pomocí lineární interpolace převádí jednotlivé vstupní hodnoty na jiné hodnoty na výstupu. Tabulky všech těchto generátorů jsou v dokumentu obsaženém na přiloženém CD (nastaveni regulace.doc).

Jako nejlepší způsob vytvoření takového generátoru se jeví jeho naprogramování v jazyce SCL a následném vygenerování funkčního bloku. Ten pouze zjistí, mezi kterými dvěma body (x_0 a x_1) se nachází vstupní hodnota a následně provede výpočet dle vzorce (3).

$$y = y_0 + (x - x_0) \frac{y_1 - y_0}{x_1 - x_0} \quad (3)$$

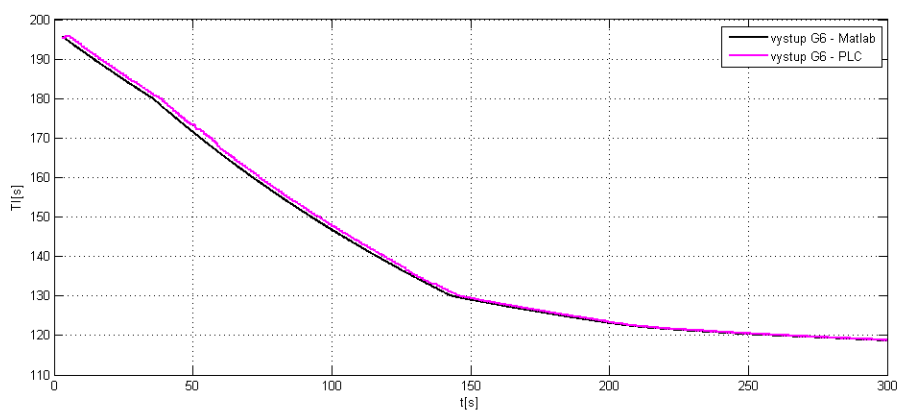
Označení funkčního bloku je FC572 a tvar jeho funkce je zobrazen v následujícím rámečku. Funkci se zadají vstupní hodnoty X a jim odpovídající výstupní hodnoty Y, přičemž je ještě potřeba zadat počet použitých hodnot.

```

// generátor G6
CALL FC 572
    Input :=DB1.DBD134
    X1    :=1.000000e+002
    Y1    :=2.500000e+002
    X2    :=3.500000e+002
    Y2    :=1.800000e+002
    X3    :=4.800000e+002
    Y3    :=1.300000e+002
    X4    :=6.000000e+002
    Y4    :=1.120000e+002
    X5    :=7.000000e+002
    Y5    :=9.000000e+001
    X6    :=0.000000e+000
    Y6    :=0.000000e+000
    X7    :=0.000000e+000
    Y7    :=0.000000e+000
    X8    :=0.000000e+000
    Y8    :=0.000000e+000
    X9    :=0.000000e+000
    Y9    :=0.000000e+000
    pocet :=5
    Output:=DB1.DBD60

```

Testováním bylo zjištěno, že se zmiňované generátory chovají téměř stejně jako ty obsažené v Matlabu. Uvedme například generátor G6 (viz rámeček výše), který je ve schématu zapojený za filtrem F1 – 200 s a zobrazen na obr. 25. Je patrné, že až na vzorkování výstupu z PLC jsou téměř totožné. Tímto grafem se také ověřil číslíkový filtr připojený ještě před generátorem G6.



Obr. 25: Porovnání výstupů generátoru G6 v Matlabu a PLC

7.3 Testování dílčích částí realizace – filtry

Protože jsem na internetu nenašel žádnou vhodnou funkční realizaci takového filtru pro PLC Siemens, vytvořil jsem nakonec svůj vlastní, který vznikl použitím Tustinovy aproximace.

Tustinova aproximace, resp. bilineární transformace, se používá k převodu mezi s-rovinou a z-rovinou. Využívá se lichoběžníkové aproximace integrálu. Tato metoda je popsána substitučním vztahem:

$$s \leftarrow \frac{2}{T_v} \left(\frac{1-z^{-1}}{1+z^{-1}} \right) = \frac{2}{T_v} \left(\frac{z-1}{z+1} \right) \quad (4)$$

$$F(s) = \frac{1}{T_s + 1} \quad (5)$$

Dosazením vztahu (4) do původního filtru (5) byl filtr převeden v z-rovině na:

$$F(z) = \frac{T_v z + 1}{(2T + T_v)z + (-2T + T_v)} \quad (6)$$

Po převodu na diferenční rovnici poté stačilo naprogramovat algoritmus počítající následující vztah:

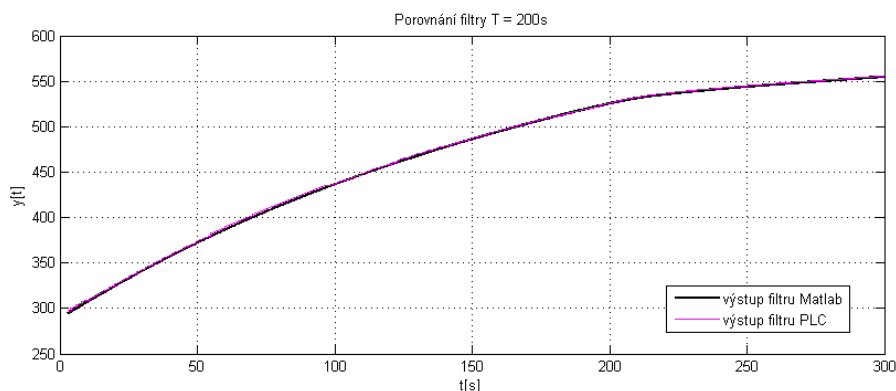
$$y = \frac{-y_{\min} [(-2T + T_v) + T_v(u + u_{\min})]}{(2T + T_v)} \quad (7)$$

Ve vztahu (7) jsou y_{\min} a u_{\min} hodnoty z minulého kroku, T_v je vzorkovací frekvence a T je časová konstanta.

Filtr byl opět naprogramován v jazyce SCL a vytvořen nový funkční blok s názvem „FC574“. Jako vstupní parametry se mu předávají vstupující hodnota, inicializační hodnota filtru, vzorkovací frekvence a časová konstanta. Dále obsahuje resetovací vstup a samozřejmě výstup.

```
CALL FB 574, DB200
Run      := DB1.DBX132.0
T        := 2.000000e+002
u        := DB1.DBD28
T        := 1.000000e-001
InitialC := 2.940000e+002
Y        := DB1.DBD134
```

Na obr. 26 je zobrazen průběh filtrované hodnoty simulované jak v Matlabu, tak i z vytvořeného funkčního bloku v PLC Siemens. Tento graf pochází přímo z běžící simulace.

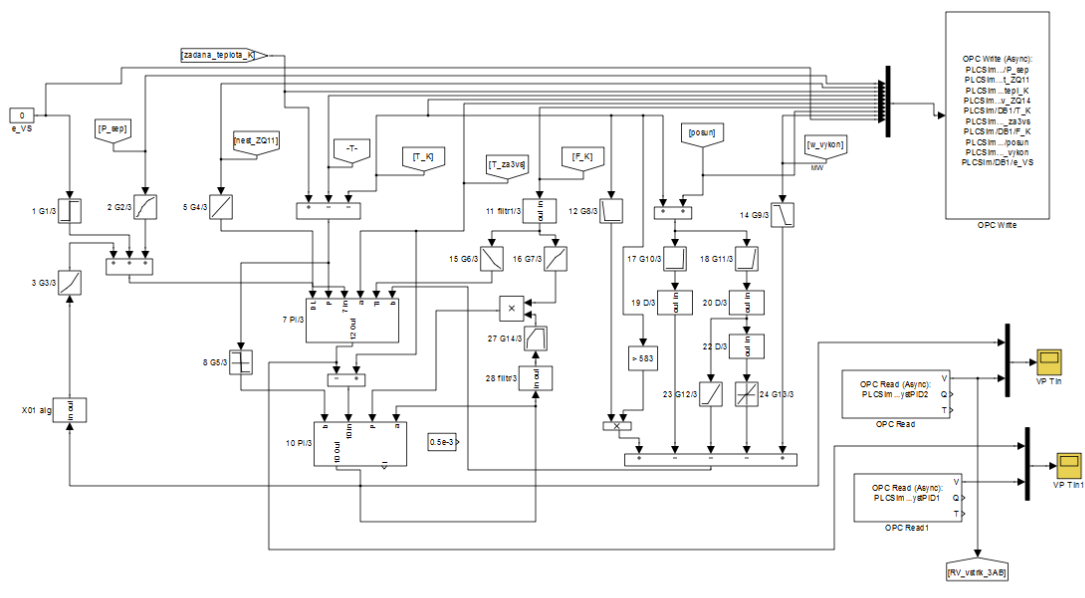


Obr. 26: Porovnání výstupů filtrů $T=200$ s v Matlabu a PLC (v běžící simulaci)

7.4 Celkové zapojení a výsledky testování

Chování PI(D) regulátoru bylo popsáno již při realizaci komunikace mezi PLC a Matlabem. Při realizaci kaskádní PI regulace dle technické dokumentace se však musely řešit další problémy. Jedním z řešených problémů bylo, jakým způsobem do regulátoru zavést počáteční podmínky. V modelu v Matlab Simulinku je toto řešeno přes „Initial condition“, kdy jsou počáteční podmínky v jednotlivých integrátozech načteny ze souboru. Po prostudování manuálu [7] jsem zjistil, že v PLC Siemens v bloku FB41 k tomu slouží přímo parametr „Initialization value of the integral action (I_ITLVAL)“. Stačí povolit použití tohoto parametru pomocí logického parametru „I_ITV_ON“. Jako nejlepší řešení načtení počátečních podmínek bylo přímo v Simulinku potřeba parametr „podržet“ na hodnotě TRUE po krátkou dobu a vzápětí jej přepnout na FALSE a vykonávat program ve standardním režimu. Obdobným způsobem je řešeno i resetování regulátoru přes parametr „COM_RST“ vždy při startu simulace. Dalším úskalím je ohlídat si správné datové typy jednotlivých parametrů regulátoru. Např. parametr k nastavení integrační časové konstanty není typu REAL, ale TIME.

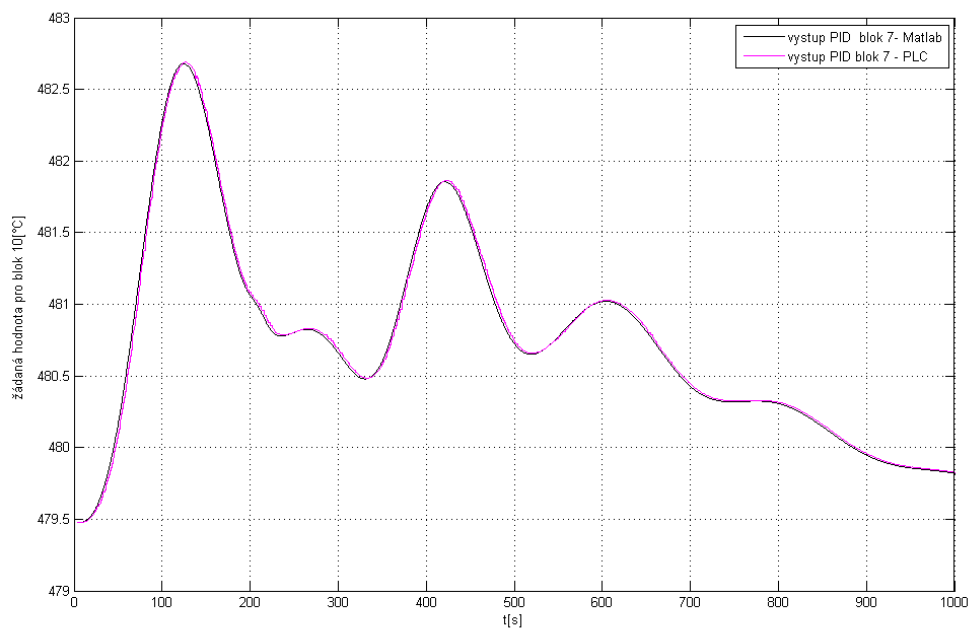
Po vyřešení všech problémů a realizaci všech dílčích částí mohla být spuštěna celková simulace včetně porovnání s řídicím systémem realizovaným v Matlabu. Schéma tohoto zapojení je na obr. 27.



Obr. 27: Řídicí část simulačního schéma s připojenou řídicí částí v PLC

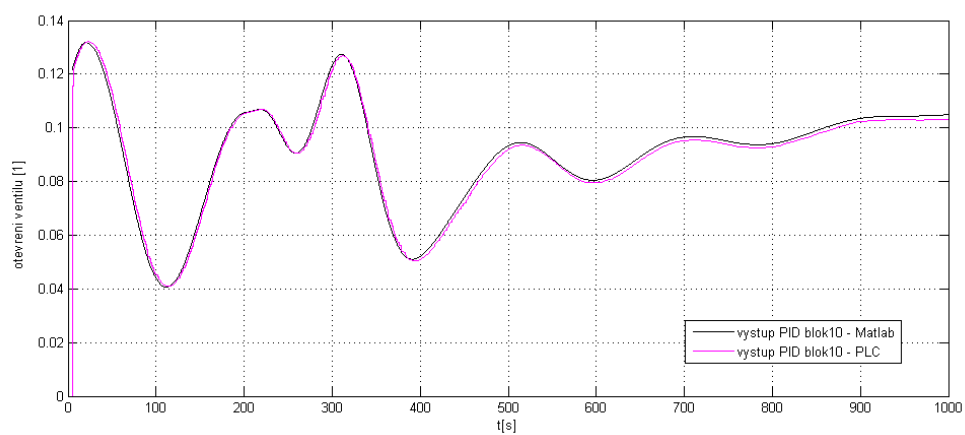
Při výsledném měření byly naměřeny charakteristiky, které porovnávají výstupy z regulátoru simulovaného v Matlabu a vytvořeného v PLC. Podle zkušeností s problémy při vytváření komunikace byla vzorkovací frekvence simulace nastavena na 1 sekundu v asynchronním režimu. V tomto režimu docházelo k celkem korektní simulaci. Simulační proces provede první skok výkonové hladiny ze 100% na 90% v čase 200 s. Samotná simulace byla testována se skoky na několika výkonových hladinách až do času 12000 s. Pro lepší názornost jsou charakteristiky na obr. 28, resp. obr. 29 zobrazeny do času 1000 s. Jak je z charakteristik patrné, tak se řídicí obvody jak v PLC, tak v Matlabu chovají velmi podobně. Díky tomu je ověřena funkčnost řídicího obvodu navrženého v PLC Siemens a tím také splněn hlavní cíl práce.

Na obr. 28 je zobrazen průběh vystupující z regulátoru vnější smyčky (blok 7 ve schéma), který zobrazuje vstup žádané hodnoty pro regulátor vnitřní smyčky (blok 10).



Obr. 28: Výstup PI regulátoru vnější smyčky

Na obr. 29 vidíte výstup z regulátoru vnitřní smyčky, který řídí již samotné otvírání ventilu. Jak je z grafu patrné, tak výstup z regulátoru v PLC je velmi podobný s výstupem regulátoru v Matlabu.



Obr. 29: Výstup PI regulátoru vnitřní smyčky

8 Závěr

Na začátku práce jsou teoreticky vysvětleny základní druhy simulací používané při metodě návrhového cyklu založené na modelu. Dále je nastíněno programování řídicího systému Siemens v prostředí Step7. Diplomová práce také představuje existující specifikace OPC technologie, které tvoří jakousi normu pro vývoj, použití a správné fungování aplikací založených na výměně dat pomocí OPC. V neposlední řadě je popsáno řízení tepelné elektrárny, konkrétně její vysokotlaké části, a je přiblížen řízený model, který je podrobněji popsán ve zdroji [1].

Práce se zabývá několika dílčími cíli, které se podařilo naplnit. Na jednodušším příkladu je ověřena funkčnost komunikace mezi PLC na jedné straně a Matlabem na straně druhé. V další části je vysvětleno vytvoření PI(D) regulátoru a řízení jednoduššího modelu reprezentovaného přenosovou charakteristikou v Matlab Simulinku. Následně je věnována pozornost popisu a testování jednotlivých částí řídicího obvodu a nakonec vyústění k řízení výstupního přehříváku pomocí kaskádního PI regulátoru sestaveného v PLC Siemens. Jednotlivé naprogramované struktury jsou porovnány s charakteristikami z Matlab Simulinku.

V průběhu tvorby diplomové práce byly vyřešeny následující problémy. Například komunikace přes OPC neprobíhala od začátku tak dobře, jak by měla, a byla použitelná od vyšší periody vzorkování, cca 1 s. Při testování vyplynulo, že odezva PLC není ideální, i když byl řídicí proces naprogramován ve smyčce 100 ms. Je možné, že mohl být problém i v použití MPI rozhraní pro komunikaci s OPC serverem. Další komplikace vycházely spíše z malých zkušeností s programováním PLC Siemens.

Hlavní snahou této práce bylo dokázat, že za celkem rozumné náklady lze vytvořit použitelný simulační model PIL. To se podařilo dokázat za předpokladu, že dané simulaci vyhovuje provoz při vyšší periodě vzorkování (např. 1s).

Seznam použité literatury

- [1] HUBKA, Lukáš. Vybrané modely funkčních podsystémů parního kotle. Disertační práce, TUL, Liberec, 2010.
- [2] JELÍNEK, Pavel. Simulace Processor In the Loop a Hardware In the Loop. Automa: časopis pro automatizační techniku [online]. Praha: FCC Public, 2007, roč. 2007, č. 05 [cit. 2012-04-12]. ISSN 1210-9592.
- [3] GOMEZ, Martin. Hardware-in-the-Loop Simulation. *Hardware-in-the-Loop Simulation* [online]. 2001 [cit. 2013-05-02]. Dostupné z: <http://www.embedded.com/design/prototyping-and-development/4024865/Hardware-in-the-Loop-Simulation>.
- [4] About Hardware in the Loop and Hardware in the look Simulation. *Opal-RT Technologies* [online]. © 2013 [cit. 2013-04-14]. Dostupné z: <http://www.opal-rt.com/about-hardware-in-the-loop-and-hardware-in-the-loop-simulation>.
- [5] HALVORSEN, Hans-Petter. Hardware-in-the-Loop Simulation. In: *Telemark University College* [online]. 2011 [cit. 2013-04-14]. Dostupné z: <http://home.hit.no/~hansha/documents/lab/Lab%20Work/HIL%20Simulation/Background/Introduction%20to%20HIL%20Simulation.pdf>.
- [6] TICHÝ, Vít. Program obnovy uhelných zdrojů v ČEZ. *Program obnovy uhelných zdrojů v ČEZ* [online]. 2007, roč. 2007, č. 4 [cit. 2013-05-09]. Dostupné z: <http://www.casopisstavebnictvi.cz/clanek.php?detail=182>
- [7] Standard Software for S7-300 and S7-400 PID Control. In: *Siemens AG* [online]. 2000 [cit. 2013-05-19]. Dostupné z: http://www.fer.unizg.hr/_download/repository/S7pidcob.pdf
- [8] STIANKO, Martin. OPC – nový standard informační technologie. *OPC – nový standard informační technologie*. 2000, roč. 2000, č. 6. Dostupné z: http://www.odbornecasopisy.cz/index.php?id_document=27739.
- [9] *OPC Foundation* [online]. ©2013 [cit. 2013-03-24]. Dostupné z: <http://www.opcfoundation.org/>

- [10] BLAŽEK, Jaroslav. OPC servery pro Simatic S7 - přehled trhu. In: *Blaja automation portal* [online]. 2012, 21.6.2012 [cit. 2013-04-19]. Dostupné z: <http://www.blaja.cz/software/opc-servery-pro-simatic-s7-prehled-trhu.html>

- [11] Product Overview Introduction (OPC Toolbox™) [online]. The MathWorks, 2010 [cit. 2013-03-21]. About OPC Toolbox Software. Dostupné z WWW: http://www.mathworks.com/help/toolbox/opc/ug/f10-6381.html#brcdx4_.

- [12] OPC Toolbox - Read and Write Data from a Model. In: *Mathworks - Documentation Center* [online]. © 1994-2013 [cit. 2013-05-15]. Dostupné z: <http://www.mathworks.com/help/opc/ug/example-reading-and-writing-data-from-the-matrikon-opc-simulation-server.html>

- [13] How can I improve the performance of a model so that I do not receive pseudo real-time violations when using OPC blocks in Simulink?. In: *Mathworks - support* [online]. 2009 [cit. 2013-05-15]. Dostupné z: <http://www.mathworks.com/support/solutions/en/data/1-2RH9OZ/index.html?product=OT>

- [14] COLONNA, Piero, Teus van der Stelt. *FluidProp software* [online]. 2004. [cit. 2013-04-12]. FluidProp v. 2.31: A program for the estimation of thermophysical properties of fluids. Dostupné z WWW: www.fluidprop.com

Seznam příloh

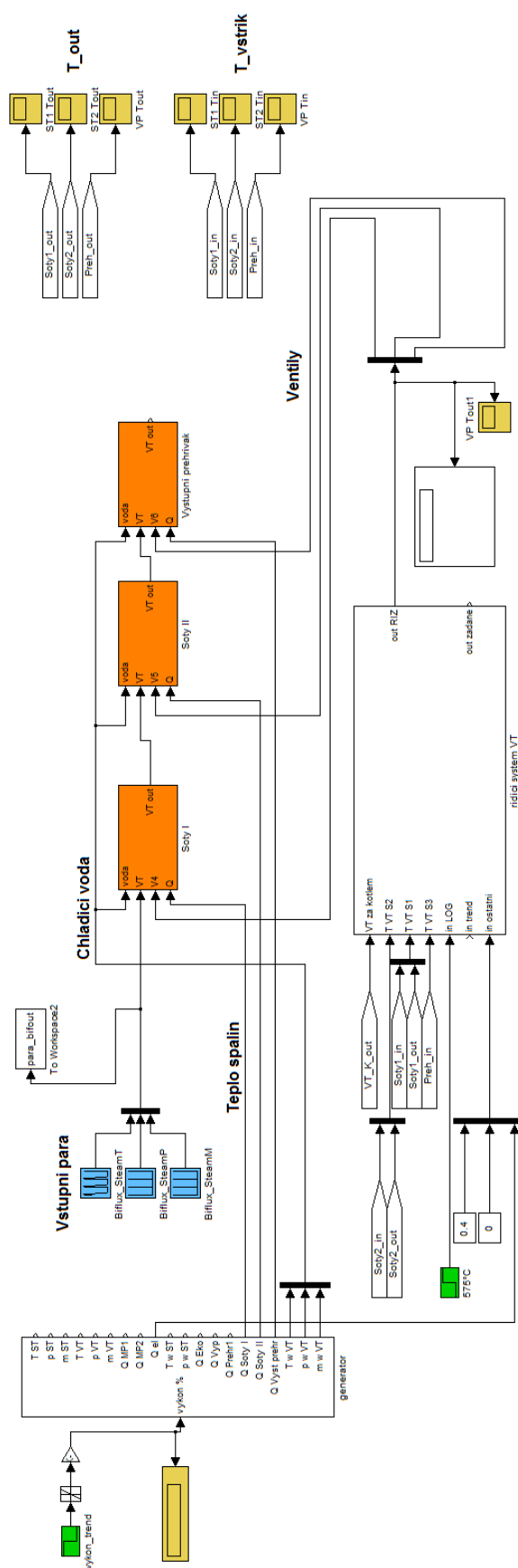
Příloha A - Schéma modelu přehřívání páry (Matlab)

Příloha B - Řídící část modelu přehřívání páry (Matlab)

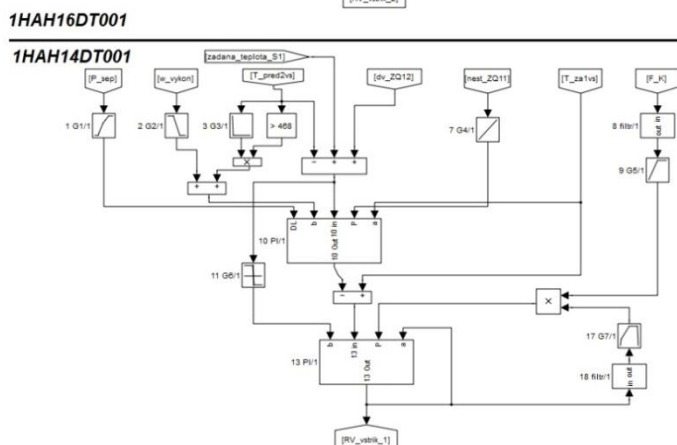
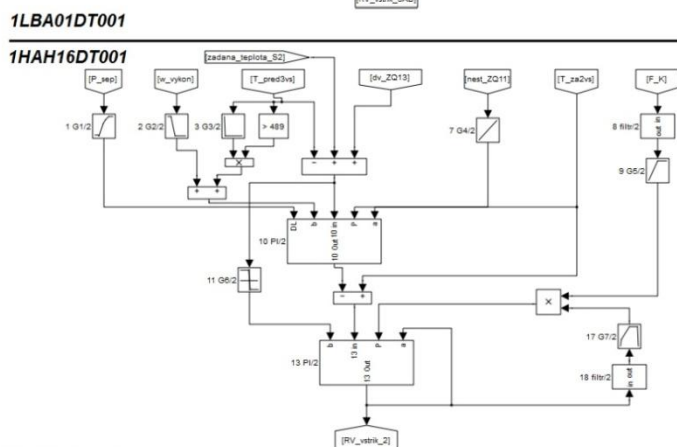
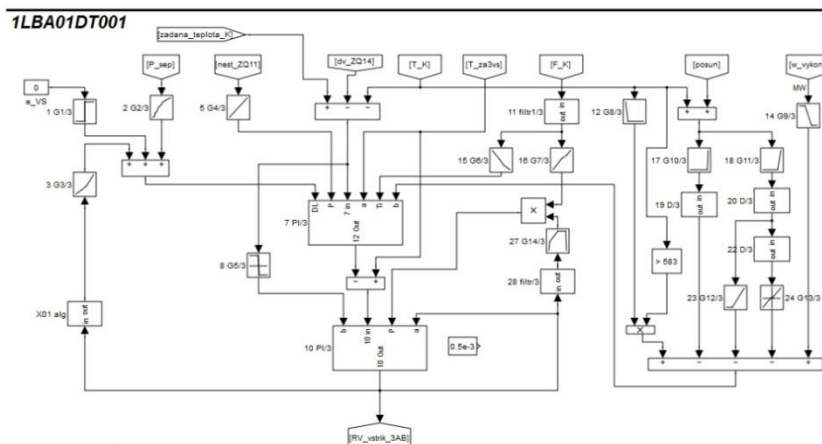
Příloha C - Instrukce ke spuštění modelu

Příloha D – Tabulka parametrů PID regulátoru FB41

Příloha A – Schéma modelu přehřívání páry



Příloha B – Řídící část modelu přehřívání páry (Matlab)



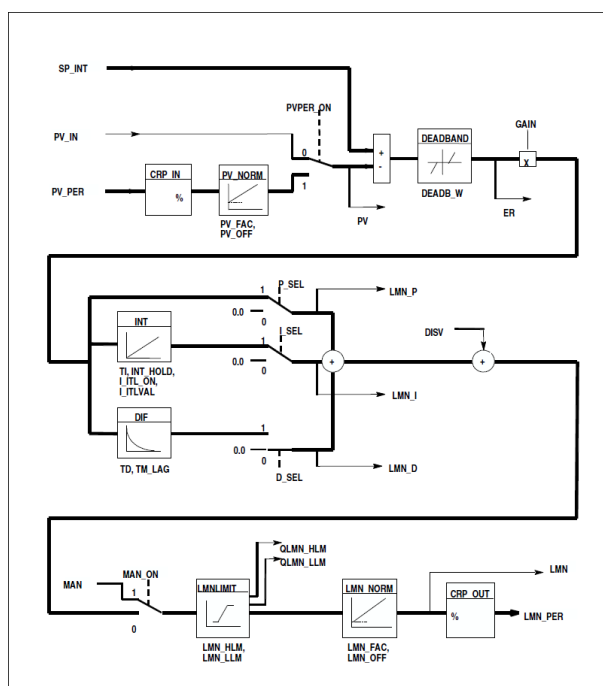
Příloha C – Instrukce ke spuštění modelu

- Pro správnou funkci modelu je potřeba nainstalovat komponentu Fluidprop 2.3 [14] obsaženou v kořenovém adresáři na přiloženém CD.
- V adresáři s Matlab soubory se nachází složka „Simulink Sample“. Touto složkou je potřeba nahradit původní složku se stejným názvem obsaženou v instalačním adresáři FluidProp.
- Při spuštění projektu v Matlabu je dále potřeba kliknout na složku „Simulink Sample“ pravým tlačítkem a vybrat: „Add to path“.
- Před prvním spuštěním modelu se musí spustit nejprve soubor „RUN2.m“, který nastaví počáteční parametry.
- Nyní by již neměl být problém spustit samotný model v Matlabu přes soubor: „Reg_VT_Cast_RO_PP1.mdl“.
- Před spuštěním modelu řízeného přes OPC server je potřeba nejprve nastavit server (viz kapitola „Propojení OPC – Siemens“). Při nastavování je vhodné nastavit stejné jméno serveru (u serveru Deltalogic se jedná při spuštění serveru: „Start Configurator -> Change -> Connection name“), jako v tomto dokumentu, tj. „PLCSim“, a proto by nemělo být potřeba měnit všechny bloky OPC Toolboxu v Matlabu. V případě použití jiného serveru, než Deltalogic či jiného názvu, je potřeba upravit všechny bloky OPC Toolbox.

Příloha D – Tabulka parametrů PID regulátoru FB41 [10]

Název	Typ Dat	Popis (Vstupní parametry)
COM_RST	BOOL	COMPLETE RESTART – Parametr sloužící k resetování regulátoru.
MAN_ON	BOOL	MANUAL VALUE ON – Aktivuje manuální režim. Defaultně nastaven na TRUE!
PVPER_ON	BOOL	PROCESS VARIABLE PERIPHERAL ON – Povolení čtení přes vstup PV_IN.
P_SEL	BOOL	PROPORTIONAL ACTION ON – Parametr ve stavu true aktivuje P složku regulátoru
I_SEL	BOOL	INTEGRAL ACTION ON – Povolení I složky regulátoru.
INT_HOLD	BOOL	PROPORTIONAL ACTION HOLD – Výstup z I regulátoru bude nastaven na konstantní hodnotě.
I_ITL_ON	BOOL	INICIALIZATION OF THE INTEGRAL ACTION – Výstup z integrátoru může být připojen na I_ITL_VAL nastavením parametru na TRUE.
D_SEL	BOOL	DERIVATE ACTION ON – Povolení D složky regulátoru.
CYCLE	TIME	SAMPLING TIME – Čas mezi voláním jednotlivých bloků PID regulátoru musí být konstantní. Hodnota udává vzorkování regulátoru (měla by být shodná s časem v OB35!).
SP_INT	REAL	INTERNAL SETPOINT – Vstup pro žádanou hodnotu.
PV_IN	REAL	PROCESS VARIABLE IN – Vstup pro regulovanou veličinu.
PV_PER	WORD	PROCESS VARIABLE PERIPHERAL – Vstup pro regulovanou veličinu čtenou z I/O periferie.
MAN	REAL	MANUAL VALUE – Akční veličina při manuálním režimu.
GAIN	REAL	PROPORTIONAL GAIN – Parametr proporcionálního zesílení K.
TI	TIME	RESET TIME – Parametr časové konstanty integrační složky I.
TD	TIME	DERIVATE TIME – Parametr časové konstanty derivační složky D.
TM_LAG	TIME	TIME LAG OF THE DERIVATE ACTION – Pokud algoritmus D složky obsahuje derivační zpoždění, lze ho přiřadit tímto parametrem.
DEADB_W	REAL	DEAD BAND WIDTH – Parametr definuje pásmo necitlivosti regulační odchylky.
LMN_HLM	REAL	MANIPULATED VALUE HIGH LIMIT – Parametr určí horní limit akční veličiny PID regulátoru.
LMN_LLM	REAL	MANIPULATED VALUE LOW LIMIT – Parametr určí dolní limit akční veličiny PID regulátoru.
PV_FAC	REAL	PROSESS VARIALBE FACTOR – Parametr udává řádový rozsah regulované veličiny na vstupu PID regulátoru.
PV_OFF	REAL	PROCESS VARIALBE OFFSET – Připočte k regulované veličině konstantní hodnotu (offset)
LMN_FAC	REAL	MANIPULATED VALUE FACTOR – Vynásobí akční veličinu zadaným faktorem.
LMN_OFF	REAL	MANIPULATED VALUE OFFSET – Připočte k akční veličině konstantní hodnotu (offset) na výstupu PID regulátoru.
I_ITLVAL	REAL	INITIALIZATION VALUE OF THE INTEGRAL ACTION – Výstup z integrátoru může být nastaven vstupem I_ITL_ON (true), tím se na výstup integrátoru dostane právě hodnota I_ITLVAL.
DISV	REAL	DISTURBANCE VARIABLE – Vstup sloužící k připojení poruchové veličiny.

Název	Typ Dat	Popis (Výstupní parametry)
LMN	REAL	MANIPULATED VALUE – Výstup z PID regulátoru - akční veličina.
LMN_PER	WORD	MANIPULATED VALUE PERIPHERAL – Normovaný výstup z PID regulátoru, je-li zpracovávána hodnota z I/O periferie.
QLMN_HLM	BOOL	HIGH LIMIT OF MANIPULATED VALUE REACHED – Pokud akční hodnota překročí horní limit, nastaví se tento parametr na TRUE.
QLMN_LLM	BOOL	HIGH LIMIT OF MANIPULATED VALUE REACHED – Pokud akční hodnota klesne pod dolní limit, nastaví se tento parametr na TRUE.
LMN_P	REAL	PROPORTIONAL COMPONENT – Výstup samostatné P složky.
LMN_I	REAL	INTEGRAL COMPONENT – Výstup samostatné I složky.
LMN_D	REAL	DERIVATE COMPONENT – Výstup samostatné D složky.
PV	REAL	PROCESS VARIABLE – Výstup sloužící ke sledování regulované veličiny.
ER	REAL	ERROR SIGNAL – Výstup sloužící ke sledování regulační odchylky.



Obr. 30: Blokové schéma regulačního bloku FB41[10]